Estimation of Bus Connection Risk with the Use of Open Bus Data

Elena Rose

University of Tampere
School of Information Sciences
Software Development M.Sc. thesis
Supervisor: Jyrki Nummenmaa
June 2016

University of Tampere
School of Information Sciences

Software Development
Elena Rose: Estimation of bus connection risk with the use of open bus data
M.Sc. thesis, 58 pages
June 2016

Key words: Bayesian binomial distribution, bus connection, journey planner, normal distribution, transfer, trip planner

Preface

Many people have contributed to this thesis by providing constructive feedback and encouragement. I want to thank my supervisor, Professor Jyrki Nummenmaa, for all his support, great ideas, and giving me the opportunity to write the thesis on such an interesting topic. Additionally, many fruitful discussions and cooperation with Paula Syrjärinne helped me shape the area of my interest and clarify the methods proposed in this thesis. I would also like to thank Jaakko Peltonen for the constructive criticism and Peter Thanisch for the valuable comments and the final proofreading. I am sure that this thesis would not exist without the contributions of Jyrki, Paula, Peter, and Jaakko.

I am also grateful to my dear husband, Josh Rose, for his love, understanding, and support during my work on the thesis.

Contents

Abstract

Bus connection risk estimation has not been studied well despite its potential impact on travellers' decisions about the choice of transportation mode and loyalty to public transportation. We aim to develop a framework to estimate and visualize bus connection chance with the use of open bus data. This thesis presents two original models for estimation of bus connection risk based on probability distributions. The first model refers to Bayesian analysis and beta distribution functions. This model depends on the possibility of calculating parameters for all possible bus connections, which is problematic since such data are not stored but rather generated during actual planning of the itinerary. The second model allows us to calculate distribution parameters for arrivals of each feeder bus at the alighting stop and departures of each connecting bus from the boarding stop. It is possible to aggregate historical open bus data to the list of distribution parameters on a regular basis, which only requires setting automatic jobs of preparing and processing data, calculating distribution parameters, and loading them to a planning graph of a trip planner. The framework consists of the theoretical description and practical application, which makes it useful for transportation systems' decision-makers, developers, researchers, and end users. The framework has been applied successfully in the city of Tampere, Finland. As a result, the web trip planner with estimation of bus connection chance is ready to use by the public.

List of Abbreviations

ANFIS	Adaptive Network-based Fuzzy Inference System
ANN	Artificial Neural Networks
API	Application Programming Interface
GPS	Global Positioning Systems
GTFS	General Transit Feed Specification
HDFS	Hadoop Distributed File System
HDI	Highest Density Interval
ITS	Intelligent Transportation Systems
MAPE	Mean Absolute Percentage Error
MFN	Multilayer Feedforward Network
PPA	Posterior Predictive Assessment
REST	Representational State
RUE	Reliability-based User Equilibrium

1. Introduction

The interest of transport administrations, city councils and software companies in Intelligent Transportation Systems (ITS) has been growing in recent times. ITS belong to a class of applications that inform end users of transportation networks how to travel in a safer, more coordinated and intelligent way [ITS action plan and directive, 2010]. The subject of this thesis is a trip planning application, which is a part of ITS helping users to plan their trips according to their conditions and circumstances. The minimum conditions that users are expected to impose on the application during the planning stage are origin, destination, date and time of departure or arrival. Additionally, some applications offer advanced searching options such as maximal walking distance, maximal number of transfers for multi-modal and multi-leg trips, prioritizing modes and routes, excluding from the search routes banned by a user etc. Based on the input information, a trip planner searches for optimal travel itineraries from origin to destination within the desired period in the graph built on the street network and vehicles' schedule information. As a result, an application generates an output with one or more alternative routes best fit to the specified conditions. Itineraries usually include step by step journey plans with textual detailed instructions and routes plotted on the map for a better visualization.

The current trend in public transport e-service is moving from implementing basic features like bus timetables, trip planners, and bus trackers to advanced functionality. Advanced functions predict future values related to the journey and estimate reliability of a journey through various prediction models based on historical and/or real-time traffic data. Consequently, we can divide predictive models into static and dynamic ones. The developers of trip planners currently prefer engaging real-time data for predicting travel times and arrivals of buses and combining static and dynamic approaches wherever possible. In the case of relying on real-time data feeds, data should contain, as a minimum, information about real-time location, speed, and pacing the schedules by vehicles but this information is not always available. Thus, real-time predictions have limitations. Dynamic predictions are possible when the buses we are interested in are already on the move, or the segment of the road on which we are measuring the speed already has some bus traffic. It limits seriously real-time systems since relevant real-time data might be absent. Sometimes there is a requirement to estimate the feasibility of the journey or the probability of timely arrival to the destination far in advance when real-time data are not available yet. Furthermore, current real-time estimation techniques focus on travel time and arrival time [Borole et al., 2013; Yu et al., 2011; Alves et al., 2012; Watkins et al., 2011; Chien et al., 2002; Mazloumi et al., 2011; Chen et al., 2013; Kumar et al., 2013; Kim and Mahmassani,

2015; Hunter et al., 2009], but they do not currently aim at predicting risks related to journeys such as a connectivity risk for multi-leg journeys.

Let us imagine that we are planning a bus trip that involves a transfer at some busy area in the city. As long as arrival of a feeder bus at the alighting stop and departure of a connecting bus from the boarding stop are timely or correlated, a transfer should be successful. However, uncorrelated lateness of buses involved in the itinerary makes the whole trip vulnerable. It is especially true if there is congestion in this area, or some other circumstances unknown to travellers impact the buses' fluency. It means that if any of two buses do not stick to their schedules, and moreover, a connecting bus can depart earlier, the risk of connection failure increases. In case of more than one transfer in the route and possible negative delays that also occur according to the recent research [Kerminen et al., 2014], the uncertainty is even larger. Negative bus delay means that a bus runs ahead of its schedule, which is likely to be the worst case for a traveller planning itineraries with changes. There are few bus stops in the city of Tampere where a bus driver has to wait for the scheduled departure time. At the majority of bus stops negative delay is not checked.

The next question is related to the amount of time between scheduled arrival of a feeder bus and scheduled departure of a connecting bus that we can consider as a safe option. One can think that two or three minutes is enough for a change, but as we can see in reality, sometimes gaps even larger than five minutes between two buses do not ensure a successful connection. A real-time-oriented trip planner can help people decrease the uncertainty by giving expected arrival times of the vehicles, but it usually does not estimate a connection risk. In addition, it cannot make real-time-based predictions much in advance of the trip, e.g. a day before the trip. Thus there is a niche for a method for trip planners to predict a bus connection risk for itineraries with transfers, regardless of the time gap between planning and actual travelling.

The objective of this thesis is to review existing models for predictions in the public transportation area applicable in trip planning applications. By comparing different approaches and studying deeply the problem of connection risk, we aim to develop a model for bus connection risk estimation based on open bus data and to implement this model in the web trip planner. The main research questions relate to the choice of relevant methods to estimate bus connection risk and to the validation of our original solution for such estimation. Besides this, the practical part of the thesis has comprised searching and studying open bus data sources, data pre-processing and mining methods, and building an application for visualization of results. The data description, the way of processing data, the model to estimate bus connection risk, and its visualization will form the original framework that can be applied by the interested parties in any geographic area in the future.

2. Predictive models in transportation

The problem of connection risk is a concern of both transportation systems' users and developers. While speculating on a connectivity problem in the transportation systems, Ceder [2007] discusses transit connectivity measures. He analyzes qualitative measures such as smoothness of transfer, availability of information channels, overall connectivity satisfaction, and quantitative measures such as average walking time for a connection, average waiting time for a connection, average travel time on a given transit mode and path, average scheduled headway, and the variance for each quantitative measure. In fact, these connectivity measures omit connectivity risk, although, in our opinion, it is a very important consideration for the benefit of all the stakeholders of the transportation system. Taking into account connectivity risk, public transport planners can produce better timetables, which is especially crucial for long and infrequent routes. Having access to connectivity risk information, travellers can leverage between their willingness to risk and travel time minimization in the journeys under planning.

Similar to Ceder, other researchers [Chandra and Quadrifoglio, 2013; Kim and Schonfeld, 2014; Muller and Furth, 2009] look into the connectivity problem from the perspective of transportation system design and coordination. Thus, Chandra and Quadrifoglio [2013] propose an analytical queuing model to find the optimal duration of the journey from the terminal, which is the inverse of a weighted sum of waiting and riding time. Having solved the scheduling problem with the use of this model, planners can optimize the connectivity and enhance the transport system performance in a given service area.

Kim and Schonfeld [2014] have developed a probabilistic optimization model for timed transfer coordination of buses. The goal of the model is to help the transportation network service manage passenger flows in a more efficient way so that transfer times of transit passengers would be reduced. In order to do it, the transportation support service should coordinate vehicles for timed transfers and headways based on the solutions found by the optimization model. Discovery of the necessary service type, vehicle size, headway, fleet size, and a number of zones has been tested successfully in the case study.

Muller and Furth [2009] have shown the positive effect of transfer planning and controlling on a traveller's waiting time. They present the term of buffer time as a component of the scheduled transfer time. The scheduled transfer time or the buffer time is the difference between the scheduled arrival of a feeder vehicle at the transfer stop and the scheduled departure time of a connecting vehicle from the transfer stop. The second component of the scheduled transfer time is the scheduled exchange time. The scheduled exchange time is the time necessary for going from the alighting stop of

a feeder vehicle to the boarding stop of a connecting vehicle. It is important to note that the buffer time is a crucial concept for planning the transfer. Increasing the buffer time raises the chance of a successful connection but results in a longer waiting time. Additionally, Muller and Furth have studied the extent to which the reliability of a transfer is improved if general operational control can reduce the deviations of arrivals and departures from the timetable.

However, even though a connectivity problem might be considered at the design stage of the transport system, existing trip planners illustrate that there is an underestimation of a connectivity problem by the journey planners' developers. Generally trip planners search for itineraries and connections based on planned timetables rather than actual bus movements. Underestimation of connectivity problem can be explained by the fact that estimating connection risk relates to decision support, and in a broader sense, to advanced functionality, which is often postponed until the core features are fully implemented. According to the study of Shoshany-Tavory et al. [2014], while engineering requirements for trip planners, public transport authorities tend to pay less attention to decision support information than to transferability, modes of transport coverage, reliability, equity, and policy support. Another explanation might be the lack of bus connection risk prediction models that can be applied in journey planners with the use of available data and without dropping the application's performance. The reasons for this limitation are, first, the fact that not all cities provide access to Global Positioning System (GPS) based bus data, and secondly, the GPS-originated bus data can constitute "big data", demanding special methods and environments for collecting, preprocessing, and processing.

Generally speaking, we can divide predictive models in transportation into ones using real-time updates and ones relying on historical data. These models are described in detail in the following sections of this chapter. The mix of historical-data-based methods with real-time data feeds is likely to provide the most accurate results in predictions. Nevertheless, a proper method should be selected with the consideration of various circumstances and tested wisely when designed for a mobile or web application.

2.1. Static models

Historical-data-based predictive models in transportation employ mainly regression analysis and probability distributions [Hans et al., 2015; Tirachini, 2013; Abdelfattah and Khan, 1998; Patnaik et al., 2004, Uno et al., 2009; Bian et al., 2015; Baptista et al, 2011; Tiesyte and Jensen, 2009; ITS Leeds, 2008; Batley and Ibanez, 2012; Lo et al, 2006; Fu et al., 2014; Syrjärinne et al., 2015; Hans et al., 2015; Grotenhuis et al., 2007;

Lian and Chen, 2013; Thanisch et al, 2014; Ng et al., 2011; Kim and Schonfeld, 2014; Hunter et al., 2009].

Regression models aim at investigating a mathematical relationship between a dependent variable and one or more independent variables or predictors. More precisely, regression analysis helps an analyst answer the question of how the changes of one independent variable impact on the dependent variable provided other independent variables are fixed. Thus, when a change among independent variables is detected, the behavior of the dependent variable can be forecast. Regression models require examining relationships between variables and finding a correct set of uncorrelated independent variables beforehand. This task demands a sufficient amount of data, a sufficient number of candidate variables, and adequate computing facilities. This, in fact, limits the application of regression models in bus data predictions. Virtually the most serious problem regarding regression analysis in the transportation field is the absence of data for many independent variables. The absolute advantage of regression models in the bus data predictions is a consideration of as many factors with a possible influence on a dependent variable as relevant data can be gathered for the study. It means that, provided sufficient data, predictions might consider all possible situations on the road that can force a bus to diverge from the schedule.

Regression models are often built to predict dependent variables such as bus delays, arrival times, travel times, and dwell times. Hans et al. [2015] have developed a method to predict dwell times by means of regression analysis. The variables included in the linear regression model are the number of alighting passengers and the number of boarding passengers. The function has been balanced by three coefficients – the average individual alighting time, boarding time, and time needed to open and close the doors. Even though dwell time might be influenced by various events like change of driver, driver breaks, early arrival, control points when departure time at the bus stop is aligned with the timetable, cash or card transactions, individual characteristics of passengers, passengers asking information, etc., it is hard to collect or simulate relevant data for all the factors. Therefore, the parameters noted above were suggested to be sufficient in the study. However, we should keep in mind that factors not considered in the model might increase significantly the dwell time.

Similarly, Tirachini [2013] has proposed dwell time estimation by means of regression models. Using a regression model similar to the model in the previous study, Tirachini has investigated dwell time's dependence on different fare collection systems, bus floor level, age of passengers, and friction between travellers boarding, alighting, and standing. Data collection has been organized as field work when an observer equipped with a stopwatch recorded data on board of buses in Sydney, Australia on weekdays for several months. In this way, the unique data including fare collection

techniques and age differentiation – school students, adults, and seniors – have been gathered. It has determined the possibility to find the impact of fare collection systems and age of passengers on dwell time. Overall, six dwell time regression models have been created. The positive effect of efficient fare collection methods, such as using prepaid cards and paying outside buses on dwell time, was discovered. Besides this, the impact of steps near the doors for alighting passengers was not statistically significant although it was proved that the steps make the boarding process slower. As expected, senior passengers increase dwell time, but the contribution of that study is not just accepting this hypothesis but quantifying dwell time differences caused by travellers' age.

Observing events on board the buses as made in the study of Tirachini is quite an expensive and time-consuming method. The restrictions regarding the amount of data and the number of variables might be overcome by simulating data rather than relying on bus probe data or observing events on board. Abdelfattah and Khan [1998] have illustrated the possibilities of the microsimulation technique in the study of bus delays. They have engaged different traffic factors in order to develop a few linear and nonlinear regression models for normal traffic conditions and for the situation when one lane was nonoperational due to road accidents or repair works. The proposed models have been validated by calibration tests and verified by field data.

Several regression models for bus travel time predictions have been created in the study of Patnaik et al. [2004]. The independent variables in the models include such factors as route distance, number of stops, dwell times, boarding and alighting passengers, time of day, day of week, trip identifier, and weather descriptors such as precipitation, visibility, and wind speed. Nevertheless, we should note at this point that the selected variables might be highly correlated, especially those related to the passenger demand. Therefore, the researchers have built several models to test various sets of independent variables separately. As a result, weather factors and weekdays have not revealed any significant effect on bus travel time. In contrast, the distance, number of stops, time of day, and dwell time have affected bus travel times considerably. Finally, Patnaik et al. argue that the models for travel time prediction restricted by trip identifier, time, and origin time independent variables are sufficient and reliable.

The conclusion made in the study of Patnaik et al. [2004], in fact, eases development of predictive models for our future research. Having studied dependencies between numerous factors, we can exclude unnecessary data from the analysis and create a more compact model, which is likely to fit into the memory of the application environment. Generally speaking, regression analysis of many variables is less frequently used in bus data predictions than probability distribution analysis of one variable due to fewer requirements to data and computational power.

Probability distributional models rely on studying statistical properties of data, checking fitness of data to a standard distribution, and estimating the distribution parameters of a variable of interest to be used at the prediction stage. A general challenge for probability distributional models applied in bus data predictions is the need to know the distribution type of data. However, sometimes data do not seem to follow any standard type of distribution. The statistic properties of bus probe data have been studied in detail by Uno et al. [2009]. The GPS-based data collected in the city of Hirakata, Japan during twelve days in December, 2003 serve as a base for the analysis. Uno et al. discovered that the observed travel times conform to the log-normal distribution in most cases although not always. Based on the findings, a methodology for evaluating travel time variability is proposed with the assumption that travel times are log-normally distributed. The methodology includes a detailed description of the gathered data, data preprocessing, data processing, and reporting of result to users.

As in the previous study, Bian et al. [2015] have attempted to describe bus data by means of a probabilistic model. The subject of the study is the service time. The service time is a sum of dwell time and time that a bus waits to enter and leave the bus stop and moves in and out of the bus stop. In fact, Bian et al. add extra time for serving a bus stop to the time that a bus spends virtually at the bus stop. Service time helps one understand whether a better coordination of a transportation network is needed. The need for coordination can be caused by queues and condensed traffic in the area of curbside stops that commonly prevail over terminal and bay-like bus stops. The proposed model deals with passengers' arrival distribution and four different scenarios for the buses approaching the bus stop. The scenarios include an empty service area, a full service area, a single bus in the first berth, and a single bus in the second berth provided two berths in bus stop area. Bian et al. have used the Monte Carlo method to estimate service time of Poisson, normal and uniform passengers' arrival distributions. The model's evaluation has shown that the Poisson distribution outperforms the normal distribution on most bus lines and has a slight advantage over the uniform distribution.

Baptista et al. [2011] have studied end-to-end travel time distributions involving travel and departure time uncertainties. In the study, each bus included in the investigated route has been tracked, and the corresponding delays have been checked for each bus stop. Then they have employed tracking information for computing conditional probabilities and modeling overall travel times from one point to another with possibly several bus transfers. The benefit of the model is the consideration of different events such as buses delayed positively or negatively, probabilities of missing buses or taking a bus out of the timetable, and dependence on time for all buses on the route chosen.

A similar approach for travel time estimation is a part of the framework proposed by Tiesyte and Jensen [2009]. The bus data gathered in Copenhagen, Denmark have been analyzed on a per-route basis with checking points at the bus stops. The trajectory data have been studied in order to discover dependencies, to find out the nature of dependencies – linear, by direction or by ranking order, and finally, to evaluate various types of predictability. Tiesyte and Jensen have classified the predictability according to the predicted values, the prediction dynamics, and the input parameters. By the predicted values predictability can be numerical or directional. Numerical predictability deals with predicting future values of travel times and arrival times, whereas directional predictability aims at predicting positive or negative direction of delays. According to the prediction dynamics, predictability is divided into static and dynamic evaluations. This classification aligns with the classification adopted in this thesis. By the input parameters, predictability can be horizontal, vertical, external, and combined. The predictability is horizontal if future values of the trajectory based on the real time trajectory measurements are predicted. It is vertical if future values of the road segment are predicted based on the historical trajectories along the route. The external predictability predicts future values based on factors external to the data and not derived from the historical travel times (e.g., weather, time of the day, and traffic conditions). At last, the combined predictability forecasts future values based on combination of vertical, horizontal, and external parameters.

The unique characteristic of the framework of Tiesyte and Jensen is a complex approach for prediction of bus travel times. While similar studies focus mainly on one type of predictability, Tiesyte and Jensen have elaborated the framework capable of embracing simultaneously numerical, directional, horizontal, vertical and external predictability. Besides this, the findings of the case study state that, firstly, the predictability of bus trajectory data is generally low, and secondly, static, and vertical predictability happen to be more reliable. In other words, according to the study, predictions based on historical data have higher accuracy than ones based on real time data.

Travel time distribution can be used not only for calculating the most probable travel time and advising travellers at the planning stage but also for estimating the reliability of the journey. In the travel reliability literature [ITS Leeds, 2008; Batley and Ibanez, 2012], the parameters of travel time distributions, which are assumed to be normal as in the many previously discussed studies, are transformed to a reliability ratio. In this case the distribution parameters – mean and standard deviation – express the expected pay-off and the inherent risk consequently. This method of travel reliability estimation is a frequently cited metric in transportation policies.

Transportation probabilities can also be estimated with Reliability-based User Equilibrium (RUE) models. RUE models relate to the concept of travel time budget, which is a sum of mean route travel and a safety margin time depending on a traveller's desired probability of on-time arrival [Lo et al., 2006]. A RUE model is based on travellers' experience in the transportation network. It assumes that all travellers desire to minimize their travel time budgets to the level correspondent to the purpose of the trip and/or individual readiness to risk a punctual arrival. Fu et al. [2014] have introduced a further development of the RUE model. In their study, multi-modal networks and, more precisely, travel time distributions consider the use of subway, auto and bus, and possible changes between all the three modes. Additionally, the parameter of fare structures is included into the model since the final cost of the trip is important for a traveller in many cases.

Apart from multi-modality, the advantage of the model is letting a traveller decide on the level of risk of being late due to including a safety margin to the calculations. The limitation of the model is the assumption that origin-destination demands, route flows, link travel times, and route travel times follow normal distributions even though the data have not been studied against the distribution. Fu et al. suggest that other types of distributions such as log-normal, Poisson, and truncated normal distributions can be adopted in their model. The other limitation is that the model has a static nature and relates to the long-term planning at the strategic level, and therefore, all travellers are supposed to have a knowledge of the traffic conditions based on their experience, that is not always the case.

Syrjärinne et al. [2015] have studied arrival time distributions with the goal of generating data-based bus timetables instead of idealistic ones, which do not take into account traffic conditions and trends of bus arrival times at a given time and area. The study proposes the statistics on bus arrivals including the earliest observed arrival times and the time span of the observed arrival times for bus customers. Practically, printed timetables are extended with the estimations of average waiting times for each bus trip at each bus stop with the use of different colors indicating three types of arrival reliability – up to four-minute waiting time, up to eight-minute waiting time and non-guaranteed arrival. Such an approach is undoubtedly novel and useful for passengers waiting at the bus stops and planning their journeys on the spot. The other interesting finding of the study gives freedom to researchers concerning the type of bus arrivals data distribution. The study illustrates that bus arrival times in the city of Tampere, Finland can be approximated as certain data sample percentiles with either normal or log-normal distribution with rather small error bounds even though the example data do not follow any standard distribution.

In contrast to many studies focusing on only one variable to be predicted, Hans et al. [2015] have attempted to construct an overall physical stochastic bus model. Their model presents a set of subsidiary models for predicting departure time, dwell time, and travel time. The data for the case study have been retrieved from the TriMet system containing quality bus data from Portland, Oregon. The other novelties of the study are, first, the model basis on analytical distributions rather than on standard distributions commonly used for such kinds of predictions, and secondly, including a new parameter – a presence of traffic signals on the links – in the travel time function. The analytical distribution of the model follows a convolution of both normal and exponential distributions, and therefore it is called normal-exponential distribution. The proposed model has been tested to reproduce empirical data, which have been further compared with the data reproduced by normal, log-normal and Gamma distributions. As a result, analytical distribution is more efficient for reproducing bus data than the other distributions because the reproduced data fit the model better in many cases with a high confidence level. This research highlights that the bus travel times in Portland are not normally distributed, which also seems to be important for future studies in bus data predictions because it evidences the need to study local data before any model can be applied to predict the data in the specific geographical area.

Despite existing discussions on reliability of travel times and arrivals, the studies on bus data probability distributions quite rarely focus on the risk of connection although many travellers are rather concerned about timely arrival at the connection stop to be able to catch connecting modes. For example, Grotenhuis et al. [2007] have found out in the survey that on-board travellers desire mostly that their remaining part of the journey go smoothly as planned, and therefore they are concerned to arrive on time at the interchanges. It is explained by the simple logic that the events of taking buses on a single route are not completely independent, because the connection can never happen if the departure of the second bus takes place earlier than the arrival of the first one. It means that, from the methodological perspective, independent estimates of two buses arriving on time are pointless. That is to say, if the connection time admitted for the change exceeds the difference between arrival time of the first bus and departure time of the second bus, the probability to follow a planned route comes to zero. As it has been stated justly by Lian and Chen [2013], the delay time for each change depends on the departure time, which is quite uncertain in reality. Therefore, the models working accurately for travel time predictions on a route without transits cannot be generalized for transit itineraries.

Thanisch et al. [2014] have investigated the risk of connection between two buses estimated based on Bayesian statistics. The prior distribution of delays of both buses at a specific bus stop computed on the data of eighty weekdays has been updated with the

ten latest real-time observations to calculate the posterior distribution. Then, they suggest using the posterior in the computation of the probability of before-deadline arrival. As in the other studies mentioned above, this model considers the arrival of buses behind or ahead of schedule, refers to historical and real-time data, and assumes that delays follow the normal distribution, but the algorithms for computing probability distributions differ.

Assessment of the probability assumes that data distribution is available and fairly accurate, although sufficient data are not always possible to get in order to find the distribution. Data might also seem to be stochastic. The study of Ng et al. [2011] addresses this problem with a distribution-free travel time model. The model requires only the first N moments of the travel time to be known and the travel times to fall to bound and known intervals. Semi-analytical probability inequalities enable one to calculate quickly upper bounds on the probability, eliminating computationally intensive methods. This model is beneficial in case of, first, having data, which do not fit commonly known distributions, and second, having performance limitations in the data processing environment.

However, it is not always desirable to operate upper bounds of the uncertainty instead of having exact probabilities at disposal. It is especially true if an interval between upper bounds happens to be too large, and therefore it leaves the investigated uncertainty still highly uncertain. For example, if this model is employed in a journey planning application, a user gets upper bounds of travel time for a particular journey from A to B instead of exact probability. If the difference between A and B exceeds some reasonable interval, the value of such a proposition for a user becomes vague. Such recommendations might make travellers feel at sea and, moreover, undermine their loyalty to the application and the public transportation system in general.

2.2. Dynamic models

As mentioned before, real-time predictions rely on dynamic models built on the basis of machine learning algorithms. The most frequently used predictive dynamic algorithms relate to the class of Artificial Neural Networks (ANN) [Seema and Sheela, 2009; Chien et al., 2002; Mazloumi et al., 2011] and travel patterns [Chen et al., 2013; Guardiola et al., 2014; Kumar et al., 2013; Kim and Mahmassani, 2015, Hunter et al., 2009].

ANN models belong to the area of machine learning. ANNs are intended to make predictions based on large amounts of data and dynamic learning of the system being supervised. Seema and Sheela [2009] have developed an ANN model with the use of seven-day GPS-based bus data collected in the city of Trivandrum, India in order to

predict bus arrival times. The data have been split to a training dataset and a validating dataset. The prediction performance has been measured by means of Mean Absolute Percentage Error (MAPE), which varies from 17 to 28% in the case study. The accuracy obtained in the case study leaves room for improvements of the model in bus arrival time predictions, which has been achieved in the following studies.

An enhanced ANN have been applied in the study of Chien et al. [2002], where ANNs integrated with an adaptive algorithm have led to a higher prediction accuracy in real time. In the case study, a 4.4-mile segment of one bus line of the New Jersey Transit Corporation provides data for the training algorithm. Due to the unavailability of GPS-based bus data in the study, the microscopic simulation system CORSIM has simulated the data. CORSIM is able to emulate bus operations including bus maneuvers and bus interactions with other vehicles competing for the road. Besides this, CORSIM is able to emulate passenger arrival distribution, which is impossible in most cases when a study is based on real data.

In the study, the data of the morning peak hours have been simulated and collected from twenty-four buses operating on the selected line. As a result of the analysis of the collected variables, Chien et al. have selected fifteen potentially explanatory variables. The variables affecting bus link travel time are bus travel distance on a link, passenger demands at stop, and average values of link volume, link speed, link delay, and queue time on a link. The variables with an effect on bus travel times from stop to stop are distance between stops; mean and standard deviation of traffic volumes, speeds, and delays; number of intersections between stops, and passenger demands at stops. Thus, two ANNs – link-based and stop-based – have been trained with different sets of the variables listed above in order to predict transit arrival times. Integration of both models to an adaptive algorithm has improved the accuracy of prediction. As a result, bus travel distance, passenger demands at stop; average link traffic volume, speed and delay in the link-based model; distance between stops; passenger demands at stop; mean of traffic volumes, delays, and speeds; and number of intersections between stops in the stop-based model have shown the smallest prediction errors. The evaluation of the proposed models illustrates high accuracy of the ANNs enhanced with the adaptive algorithm in bus arrival time predictions.

Mazloumi et al. [2011] have proposed the integrated framework with two ANNs to predict the average and variance of travel times. They have collected test data from one bus line in Melbourne, Australia about an eight km long route for a six-month period. Bus schedule adherence data have been combined with degree-of-saturation data being received from inductive loops of the intersections on the route in the latest fifteen minutes before the departure of a bus from the point timed. The combination of these variables aims at dynamic responding of the predictive model to the changes in the

traffic. However, the model combined with real-time data has revealed a minor improvement of the predictive accuracy of ANN. To make a conclusion, historical-data-based models present an easier and fairly reliable option for predicting bus travel time.

The problem related to applying ANN-based models in web and mobile applications is the requirements for the application environment to operate with relatively small datasets input for processing and for fast methods of processing. In the studies proposing the ANN model for predicting bus journeys only one bus route is usually selected as a test bed. The reason for this is that coverage of all the routes of the city increases drastically the complexity of ANN. Such heavy computations require computing capacities too enormous for web and mobile applications. It leads to the conclusion that, if data mining algorithms are used for predictions in transportation, firstly, the predicting stage should take far less time than the training stage, secondly, the training stage should occur in a powerful computational system separated from end-user application. A separate computing system can transfer only parameters of the model to the productive application.

Travel patterns present groups of similar travel trajectories measured in temporal and spatial dimensions. Identification of travel patterns yearly, monthly, weekly, daily or hourly enables one to impose once found patterns on the real-time traffic situation and define behavior of the travel time, arrival time, speed or delay uncertainty according to the pattern. Chen et al. [2013] have studied traffic speed patterns for a road link with the use of two soft computing models – the Multilayer Feedforward Network (MFN) and the Adaptive Network-based Fuzzy Inference System (ANFIS). They have tested the models on Beijing's urban probe vehicles data in order to check the models' robustness to the missing data, which is highly probable with probe data, and the models' generalization capabilities. They have found out that ANFIS offers a better model of traffic trends in studied segments than MFN, helps one discover meaningful hidden traffic speed patterns, and it is utterly robust to missing data.

Even though travel patterns are frequently used in dynamic models, they can serve as a basis for static models too [Guardiola et al., 2014; Kumar et al., 2013]. Guardiola et al. [2014] have researched the daily traffic flow profiles with the use of functional data analysis based on historical data. The study proposes to build multivariate flow charts based on historical data captured during one or more years. Then these charts can serve for monitoring shifts in traffic profiles in the future providing the meaningful information for decision-makers, e.g. the need to add extra lanes in the highways. The requirement of the model is preprocessing one-year historical data to remove outliers and build control charts describing the stable condition. This is the weakness of the model since one-year data is a large quantity in comparison with what we needed for

other techniques. Furthermore, such a quantity of data is not always available because, for example, the timetable changes more frequently.

Kumar et al. [2013] have analyzed GPS-based bus data separately for each day of the week to obtain weekly, daily and time-wise patterns. The analysis has covered one route in the city of Chennai with fourteen trips per day during two months. They have split the data to 100-meter portions in the final sample for each trip. As a result, they have discovered similar travel time patterns for all days except Sunday. The most important issue of applying the described frameworks in travel time predictions is their dependence on the location, and the need for data-intensive calculations. One should investigate travel patterns behavior in specific geographical areas where the predictions are going to be carried out.

Obligatory binding to location has been overcome by Kim and Mahmassani [2015]. They have proposed an original trajectory clustering method to discover travel patterns in a traffic network. At first, they have identified spatially distinct traffic flow groups using trajectory clustering, and then they have investigated each spatial group to discover temporal patterns. The framework is supposed to be applicable in any road network without the map-matching preprocessing step. Data processing includes similarity measurement, trajectory clustering, generation of cluster representative subsequences, and classification of trajectories. The trajectory clustering method has been tested successfully on actual traffic data collected from New York City, New York. A simple experiment has illustrated the possibility of application of the framework in the network-level traffic flow pattern analysis and travel time reliability analysis.

Hunter at al. [2009] have presented a combination of travel time distributions and travel pattern methods. GPS data from probe vehicles gathered in San Francisco, California have enabled them to build a probabilistic model of travel times through the arterial network. Then they have used an expectation maximization algorithm for learning the parameters of the probabilistic model. Finally, they have extended the model to the unknown parts of the transport network. Hunter at al. have learnt general traffic patterns of each day of the week at each time for a transport network and save them in a short, summarized form. The transport network has been represented as a graph consisting of directed links. Each link is characterized by the set of parameters: the length of the link, the number of lanes, the presence of traffic lights, congestion on the given and neighbor links. In addition, temporary conditions such as weather or sport events are considered as factors able to change a typical behavior or patterns in the link. Hunter et al. employed Bayesian inference for building a probabilistic model with the assumption that travel times data follow normal or log-normal distribution. The goal of the study was to find historical travel patterns for building a real-time model in the future. A real-time model is expected to be updated continuously with estimates

obtained from real-time incoming data in order to predict traffic conditions. In other words, traffic patterns should help one deal with limited streaming data due to the lack of probe vehicles or losing connections, which is often the case in gathering real-time data.

A challenge for travel pattern methods is the requirement to process a large dataset that can be unavailable. Processing facilities can also be insufficiently powerful. Thus, similarly to ANNs, travel pattern methods assume difficulties not only at the analysis stage, but also at the storage and retrieval stage when enough disk space and operational memory must be allocated for analyzing and predicting applications due to large datasets to be input. Therefore, applying such methods in an online trip planner challenges the performance.

The other serious problem of dynamic models is a possible absence of real-time data relevant for predictions at the time of a user's request. That is to say, real-time data ease short-term planning better responding to dynamic traffic conditions but these data are not applicable for long-term planning. In the case when real-time data cannot serve as a base for predictions due to absence of relevant data at the time of trip being planned, predictions can use historical data.

All the models discussed above have the potential to be implemented in web and mobile trip planners. The choice of the probability distribution model developed further in the thesis is explained by its relative simplicity and reliable results. In addition, a compact form of the final data to be loaded into the web server's memory is the other advantage. While traffic patterns require a large amount of data to be processed for each user's request in order to give reliable advice, the distribution parameters are much more compact. Keeping in mind that big data cannot be processed fast enough to keep the performance of trip planners reasonable for online service, a trip planner's developer has to select a predictive model and design properly a whole system. A few-second response time is the basic requirement for online applications. The findings of the previously discussed studies [Mazloumi et al., 2011; Tiesyte and Jensen, 2009] that historical-data based models suggest the same or even higher accuracy than real-time-data based models, support our choice of the method related to probability distributions. Putting into practice the framework that we propose, it is possible to develop a trip planner able to estimate a risk of bus connection online.

3. Data

We exploit several sources of open public transportation and map data in this thesis. Nowadays many municipalities tend to open public transportation data for common usage in order to attract the attention of interested parties to problems and areas to be improved in transportation information. At the least, opening data initiates the interest of software developers and scientists to apply different methods to the data and develop applications to solve existing problems. Consequently, it leads to a rise in the willingness of the public to use the product produced in the field in question. Thus, opening bus data increases the potential of a larger usage of buses due to the appearance of new applications and the elaboration of new methods in transportation planning, which improve navigation and travelling.

The city of Tampere has opened its bus data long ago, and it is currently aiming at opening more transportation data. Due to the availability of such data, we are able to experiment with different methods and propose models, which can improve the transportation situation in Tampere. Thus, in this work, we use a few sources of open data. First, real-time bus movement data can be obtained through the Journeys Application Programming Interface (API) [Journeys API]. Journeys API allows developers and clients to access real-time one-per-second bus location information in the region of Tampere via the Representational State (REST) API. Secondly, the latest bus timetables and routes are provided on a regular basis by ITS Factory in General Transit Feed Specification (GTFS) files [GTFS for Tampere] formatted in accordance with the GTFS industrial Google standard. Lastly, Open Street Map [Open Street Map] data feeds are essential for applications based on map visualization.

3.1. Journeys API

The source of the data in Journeys API is derived from GPS trackers installed in all the buses operating in the city of Tampere. There are a few APIs distributing these data openly, but at the moment of the study we selected the most recent one, Journeys API. In general, there are more data items in this API than we require for the purposes of the application developed as a part of this thesis. Besides dynamic bus data feeds, the API provides static information about routes, lines, journey patterns, journeys, bus stops, and municipalities. As far as static data are relatively constant, we poll Journeys API every second for only dynamic vehicle activity data to be stored for further analysis. In case of the need for static data, we send requests to the API directly during the programs' execution. The raw data that we collect contain the list of elements described in Table I.

Table I. Raw bus data

Name	Type	Meaning
1	2	3
Time	Time stamp	It is a combined date and time in UTC expressed according to ISO 8601 in the format "YYYY-MM-DDThh:mm:ss.Ms+hh:mm". It specifies the point of time when the vehicle's activity is monitored. E.g. "2014-11-27T14:18:19.020+02:00" is 14:18:19 November 11, 2014, +02:00 time zone.
LineRef	integer	It indicates the line number. A letter in the line number is removed if exists.
DirectionRef	integer [1, 2]	On any given bus line, a bus can be travelling in one or two directions. The bus company assigns a number, "1" or "2", to each of these directions. E.g. Line 26 has Direction 1 from Höytämö to Kaarila and Direction 2 from Kaarila to Höytämö in the city of Tampere.
DataFrameRef	date	It specifies the date in the format "YYYY-MM-DD" when the vehicle started from the origin stop.
Latitude	double	It specifies the bus's latitude coordinate in decimal degrees at the time of observation.
Longitude	double	It specifies the bus's longitude coordinate in decimal degrees at the time of observation.
OperatorRef	string	It specifies the name of the bus operator.
Bearing	integer	It specifies the azimuth angle of the bus. It is equal to zero if the bus is stationary.
Delay	integer	It specifies the amount of seconds the bus is delayed from its scheduled timetable. It is negative if the bus is ahead of its schedule.
VehicleRef	string	It identifies uniquely the monitored vehicle. However, this field is empty quite often.
JourneyPatternRef	string	It indicates the line number with possible letters. Generally, line numbers consist of only numbers, but sometimes they might contain a letter in the line name indicating small differences in the routes in comparison to the main route (e.g. 9K).
OriginShortName	string [4]	It specifies the origin stop number where the vehicle started the journey.
DestinationShortName	string [4]	It specifies the last stop number in the journey.
OriginAimedDepartureTime	string [4]	It specifies the departure time from the origin bus stop in the format "hhmm".
Speed	double	It indicates the vehicle's current speed in km/h.
TimeAPI	time stamp	It is Epoch Unix time stamp indicating a number of seconds from the Epoch start until the current time. The time of day is in Universal Time Coordinates, so it must be adjusted by two hours to convert to Finnish time.
TimeStorage	time stamp	It is Epoch Unix timestamp indicating the number of seconds to the moment of receiving the data by the server. It can be used together with "TimeAPI" to calculate the delay from data generating to data receiving.

We save real-time data every hour to separate comma-separated values (CSV) files in order to collect sufficient historical data for further analysis and experiments. On average, the daily data size amounts to about 650 Mb. It is larger on weekdays and smaller on Saturdays, Sundays, and public holidays due to a smaller number of buses in operation. It can be noticed in Table I that the data only give the location of the specific bus at the specific time but not the arrivals or departures at the bus stops. It means that the raw data have to be pre-processed before they can be applied in the models for arrival times and departure times estimations.

Furthermore, there are some known issues about real-time bus data in Tampere that are highlighted in previous studies [Syrjärinne et al., 2014; Kerminen et al., 2014]. It requires data cleaning before the data are going to be analyzed. Data cleaning should address properly duplicates, missing data, and erroneous records, which can be caused by malfunctioning transmitters, lost connection and other technical problems.

3.2. GTFS

The GTFS standard defines the format for bus timetables and static location details on bus routes. The GTFS format lists different properties of a bus transportation network in a predefined structure. However, it is the decision of a data provider what properties from the full list will be provided. The GTFS data of the city of Tampere are open and updated normally twice a year when there are changes in bus timetables according to the summer or winter mode. Additionally, when the city of Tampere issues new rules on lines and routes, the GTFS files are updated accordingly. The GTFS files of Tampere contain data about bus agencies, bus stop locations, routes, stop times, and calendar. The full description of GTFS provided by the city of Tampere is presented in Table II. A specific bus route characterized by line number and origin departure time is uniquely specified by a ten-digit identifier “tripID”.

There are two problems related to the integration of two sources of data – the GTFS and Journeys API’s data – in one application. First, real-time bus data do not contain a unique trip identifier “tripID”. Secondly, a line is frequently specified by only a number whereas GTFS identifies the same route as a combination of a number and a letter. A letter in a line number means that the trip can differ slightly from the basic route (e.g. it can include an extra bus stop or a few bus stops or run along different streets in one or a few segments). Even though there is a special entity “JourneyPatternRef” in Journeys API, which is supposed to present a line as a combination of a number and a letter (see Table I), in practice there are very few records where the “JourneyPatternRef” element contains a letter. The unique identification of the trip in real-time bus data can be provided only by means of a

composed key consisting of a line, origin bus stop, destination bus stop and origin departure time. Consequently, when one tries to integrate these two sources of data, he or she will face trips found in GTFS, which are impossible to relate correctly to real-time data.

3.3. Data pre-processing

Data cleaning and preprocessing steps require reading the whole data and fulfilling different operations such as sorting, ordering, grouping, and searching. Having historical data of about 650 Mb per day, we need a powerful computing system capable to deal with big data. It is especially true in the case of analysis based on data gathered during a long period of time.

At the cleaning step, we group the data by a composed trip identifier, discussed above, and sort them within each group. Then we select only the correct and full records related to the trip as an output. At the actual preprocessing step, the real-time data should be combined with a sequence of bus stops identified in another interface of Journeys API or in GTFS files. Our algorithm searches for the sequences of bus stops and bus stops' coordinates in Journeys API by a composite key consisting of the line, origin code, destination code and origin aimed departure time. The request string for retrieving these data follows a template:

http://data.itsfactory.fi/journeys/api/1/journeys/[line]_[origin_aimed_departure_time]_[destination]_[origin],

where "[line]" is a full line number with letters if they exist, "[origin_aimed_departure_time]" is a time in the format "HHmm" when the journey starts, "[destination]" is a code of the destination bus stop, and "[origin]" is a code of the origin bus stop. The time of the journey start can be found in the GTFS files. We should mention at this point that the response to the requests might be empty due to technical problems on the provider's side. For example, there is an empty response for line 41 starting at 13:10 from the bus stop 8052 and running to the destination bus stop 8024. If we attempt the request string http://data.itsfactory.fi/journeys/api/1/journeys/41_1310_8052_8024, the response is empty even though there are real-time bus movements' data for this journey. Our algorithm discards the whole journey in such cases.

Table II. GTFS content provided in the city of Tampere

File name	Element	Example
1	2	3
Agency	agency_id	JOLI
	agency_name	Tampereen joukkoliikenne
	agency_url	http://joukkoliikenne.tampere.fi
	agency_timezone	Europe/Helsinki
	agency_lang	fi
	agency_phone	+358356564700
Calendar	service_id	TAL_AR_K28_2016
	monday,tuesday,wednesday,thursday,friday,saturday,sunday	1,1,1,1,1,0,0
	start_date	20150810
	end_date	20160605
Calendar_dates	service_id	TAL_AR_K28_2016
	date	20150811
	exception_type	2
Routes	route_id	1A
	route_short_name	1A
	route_long_name	Vatiala - Pirkkala
	route_type	3
Shapes	shape_id	1325105147016
	shape_pt_lat	61.49733
	shape_pt_lon	23.76612
	shape_pt_sequence	1
Stop_times	trip_id	4530743642
	arrival_time	11:05:00
	departure_time	11:05:00
	stop_id	0031
	stop_sequence	1
Stops	stop_id	0001
	stop_code	0001
	stop_name	Keskustori M
	stop_lat	61.49751
	stop_lon	23.76151
Transfers	from_stop_id	5217
	to_stop_id	5217
	transfer_type	2
	min_transfer_time	1
Trips	route_id	42
	service_id	TAL_AR_K28_2016
	trip_id	4530743642
	trip_headsign	Tampere
	direction_id	1
	shape_id	1317200319250
	wheelchair_accessible	0

After extracting the sequence of bus stops with longitudes and latitudes for each journey, we define a fifty-meter radius area around each bus stop to track vehicles in these areas. The value of fifty meters for the radius should be adjusted according to actual physical traits of bus stops in a city. The radius definition is necessary because in practice it is quite difficult to identify the exact time of arrival or departure. First, the type of a bus stop should be kept in mind (e.g., curb side, bay- or terminal-like bus stops) but most likely there will be no data about the types of all the bus stops in the open data source. GTFS does not specify the type of bus stops. Neither Journeys API providing bus data in Tampere does it. Second, buses can bunch up, build quite a long line near the physical bus stop, and consequently open the doors for boarding and alighting quite far from the point that we would assume as a bus stop. It challenges the stop area definition and requires defining some area instead of a point as a bus stop. Last but not least, even though there are data about bus speed, we cannot consider zero speed as a moment when a bus is located at the bus stop. The reason is that buses can pass by the bus stop if there are no requests to stop, and, furthermore, they can stop in front of the intersections close to the bus stops.

Bearing in mind these restrictions and experimenting with different radius values, the fifty-meter radius is chosen as an optimal value. After radius defining, bus positioning data are scanned to look for arrivals and departures for each bus stop found in the bus stop sequence for each journey. We consider the minimum time for each trip identifier within one bus stop as a vehicle's arrival time. Similarly, the maximum time within a defined area of a bus stop is a vehicle's departure time. Our algorithm is based on the algorithm of offline computation of link travel times proposed by Syrjärinne and Nummenmaa [2015]. The input, simplified pseudocode, and description of the functions of our preprocessing algorithm for arrival and departure time computations are listed in Table III.

As a result of data cleaning and preprocessing, the raw bus data are cleaned and aggregated to the output described in Table IV. The elements "Line" and "Direction" are not compulsory but might help one understand better the data. The compulsory elements of the output are "Journey Pattern", "OriginShortName", "DestinationShortName", and "OriginAimedDepartureTime" serving as a compound key for a trip identification. Besides this, "StopCode" is indicating the code of bus stops. The calculated values of arrival time "ArrivalTime" and departure time "DepartureTime" are necessary for the data analysis. In other words, we summarize the data in the form where each trip identifier contains only arrival and departure times at and from the bus stops in the sequence determined by the trip's route.

Table III. Preprocessing algorithm for arrivals and departures computations

Input	Pseudocode	Description of functions

Data is one-day historical bus data	*G = Map(Data)*
	for*(each g in G)*
	stops = ScanAPI(g)
	arrivalTimes.add(g.time)
	for*(i=1:length(stops))*
	s = stops[i]
	for*(j=1:length(g))*
	d = FindDistance(s.position, g.position)
	if*(d <= Radius)*
	ArrivalTime=min(arrivalTimes[j])
	DepartureTime=max(arrivalTimes[j])
	endif
	endfor
	endfor
	endfor
		Map is a "mapper" function that groups and sorts data according to a defined key.
		ScanAPI is a function to request Journeys API for a sequence of bus stops with their coordinates according to a key formed in a mapper function.
		FindDistance is a function calculating the distance in meters between two points expressed in a pair of longitude and latitude.

For the efficiency of data analysis we process raw data on a daily basis in order to form an aggregated CSV file with arrival and departure times of the previous day. This daily procedure enables us to fulfil fast data analysis since the data are summarized and decreased in size significantly. The data can be taken from any period of time, and the need to process the same piece of raw data again disappears while data analysis is done.

The framework we have chosen for data cleaning and preprocessing is the MapReduce programming model elaborated by Apache. MapReduce is a programming framework for parallel processing of big data in a distributed system. Java libraries of MapReduce are used to program the algorithms of preprocessing in Java and execute them in the distributed cluster run under Apache Hadoop. The framework's main components are the "mapper" and "reducer" functions. At first, the mapper function processes input data sequentially line by line to form pairs of a key and value, which can be of any standard or programmed type. Then the data are sorted in an ascending order by key. After that, the result of the mapper function is transferred to the reducer function which merges all values associated with the same key in a way programmed by

a developer. MapReduce offers an easy-to-use and powerful interface for forming key and value pairs and coding mapping and reducing algorithms for large datasets.

Table IV. The format of preprocessed data.

Name	Type	Meaning
1	2	3
Line	integer	It indicates the line number. A letter in the line number is removed if exists.
JourneyPattern	string	It indicates the line number with possible letters (see Table I).
Direction	integer [1, 2]	It specifies the direction the bus is travelling in (see Table I).
OriginShortName	string [4]	It specifies the origin stop number where the vehicle started the journey.
DestinationShortName	string [4]	It specifies the destination stop number where the vehicle is heading to.
OriginAimedDepartureTime	string [4]	It specifies the departure time from the origin bus stop in the format "hhmm".
StopCode	string [4]	It specifies the code of the bus stop.
ArrivalTime	time stamp	It is Epoch Unix timestamp indicating the number of seconds to the moment when the vehicle appeared first in the area of the bus stop.
DepartureTime	time stamp	It is Epoch Unix timestamp indicating the number of seconds to the moment when the vehicle disappeared from the area of the bus stop.

The MapReduce program is automatically parallelized in a cluster of many machines, which guarantees processing big data quickly. The framework itself solves many problems related to parallel programming. Thus, a developer of a program is freed from the requirements to partition input data, schedule execution, and manage machines failures and inter-communication by a developer. The only requirements left are to design keys and values and configure the cluster properly.

In the thesis experiments, we deployed the Hadoop MapReduce system under the Cloudera platform distribution on a distributed cluster of one master machine and two slaves. The master is an eight-core machine with 80 Gb of disk space. The remaining two machines operate as slaves (or data nodes in terms of Hadoop). Every slave

operates with twelve cores and 7.2 Tb disk space. Historical bus data are saved hourly as CSV files to the Hadoop Distributed File System (HDFS) of the cluster. Storing data in HDFS is a requirement of MapReduce framework for input data. It creates the only data input point and enables a MapReduce program access data from any machine in the cluster.

For the timely update of the content of our final application, our cluster executes programs for data cleaning and processing on a daily basis to find arrival and departure times for each trip identifier and each bus stop in the sequence for the route of the trip as was discussed above. Fig. 1 illustrates the logic and key-value design of the MapReduce task doing data preprocessing. This MapReduce program runs automatically every night at 3 a.m. to process the raw data of the previous day.

Hadoop allows a developer to gather all statistics of MapReduce jobs, which have ever been run in the cluster. According to the statistics, the running time of the preprocessing task is six minutes on average. As a result of the proposed design, bus data in a set of twenty four files with an overall size of around 650 Mb are shrunk to one file with a size around 5Mb every day. Expressing the result as the amount of data that has to be analyzed further, we get around 130 times less data at the analysis stage than we would have without cleaning and preprocessing. Although the initial amount of data is not very large, we use data processing because it is more scalable. This is especially important if data gathered during a large period of time in a big city are needed for the analysis. All in all, the larger the period of history we want to include to the calculations, the more sensitive to the data size a program will be.

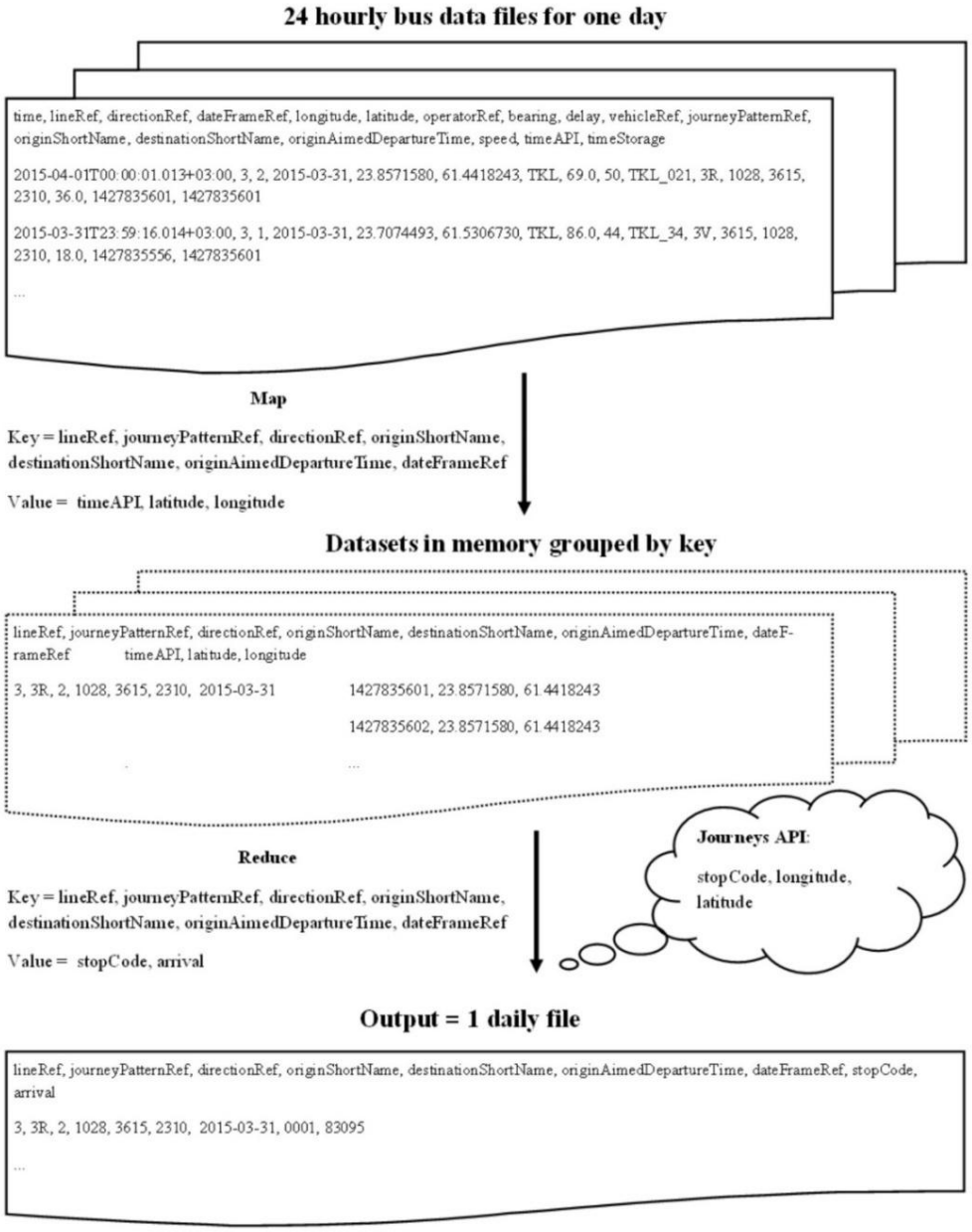

Figure 1. The MapReduce task for getting arrival and departure times data.

4. Methods

We have compared different models and methods most frequently used in predictions for transportation area in Chapter 2. The comparison of the models revealed the fact that, despite the popularity of dynamic models and predictions based on real-time models, static models are easier to implement in applications. The advantage of static models is the opportunity to process historical data in advance in a separate system, thereby avoiding adverse effect on the performance of the end user application. Furthermore, it was established that methods belonging to static models are as reliable and accurate as, and in many cases even more accurate than, methods associated with dynamic models. Therefore we decided to construct our framework for bus connection prediction on the historical-data-based methods. In this chapter we present two original frameworks, which are expressed through probability distributions. Both frameworks are original and novel for transportation systems although their cores are based on well-known statistical methods.

4.1. Bayesian binomial distribution

It is common for a scientific hypothesis to be expressed in terms of a probability distribution to which the conformance of observable data can be measured. The distinct characteristics of the particular distribution are defined by the parameters, which determine the shape of a probability distribution curve. As is typically the case, the values of these parameters are our unknown variables. Previous knowledge about parameters is expressed in Bayesian analysis as the prior distribution or, simply, the prior. Knowing the prior and observing new data, Bayesian analysis enables us to build a new probability distribution, called the posterior distribution or the posterior, which can be used for prediction of future values of observable data.

A connection between two buses may or may not be successful. This simple statement allows us to deal with binomial data expressing mutually exclusive events. Bayesian analysis for binomial data is mathematically convenient to perform based on the beta density function [Kruschke, 2011; Penttinen and Piche, 2010]. Transformation of data from a nominal form to a binomial one gives us freedom in connection risk estimation since binomial data do not require knowledge about the distribution type of the original data. Furthermore, the beta distribution function describing binomial data is quite easy to compute.

4.1.1. Bayesian binomial analysis

In order to present our framework for bus connection estimation in detail, we take one itinerary with an interchange proposed by the journey planner of Tampere [Tampere Journey Planner] as an example. In the study of Syrjärinne et al. [2014] on bus movements in Tampere, the risk of delay was shown to rise during rush hours on weekdays, especially for long routes. It determines our choice of the itinerary, which includes two long route lines and a transfer in the city centre. The itinerary goes from the West of the city to the East and requires a change from line 26 to line 27 at the bus stop Tipotie, which is located almost in the very centre of the city. The arrival times of lines 26 and 27 at this bus stop are scheduled consequently at 08:13 and 08:16. Both lines cross the city from the North-West to the South-East via its central part, and this morning hour is subject to traffic congestions. Fig. 2 illustrates the density curves of arrival times and departure times of the vehicles in question at Tipotie in March 2015. The algorithm for density estimation in this figure computes Gaussian kernel density estimates with the given observations of arrival times.

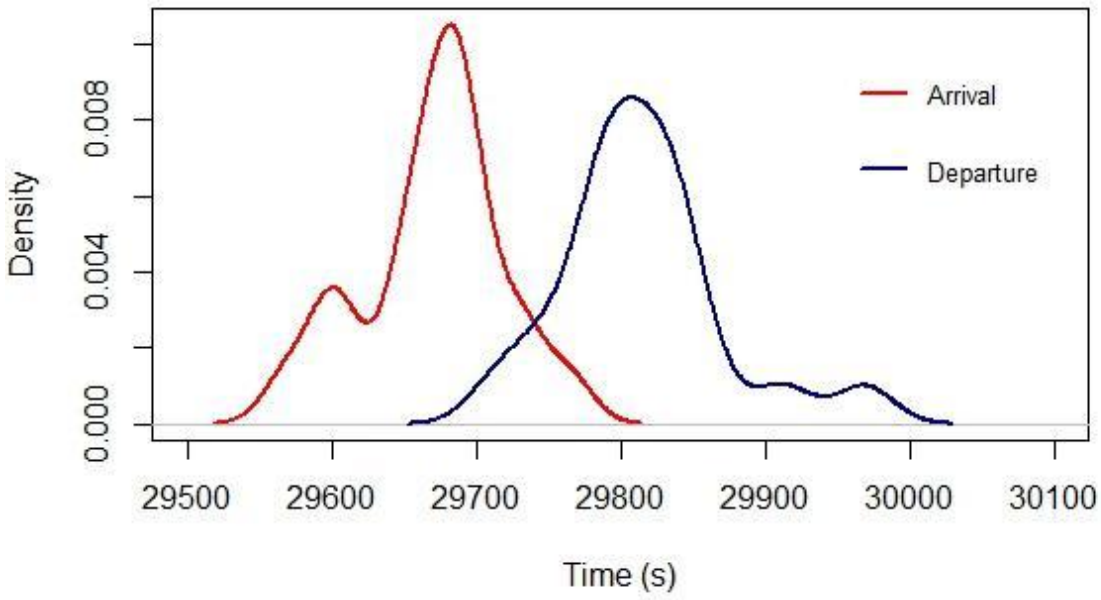

Figure 2. Arrival times of line 26 scheduled at 8:13 and departure times of line 27 scheduled at 8:16 on weekdays of March 2015

The change bus stop is the same for both lines, so that one-minute minimum transfer time between arrival of line 26 and departure of line 27 should be sufficient for successful connection. According to the timetable, there is a three-minute gap between the vehicles' arrivals. Hence, a user has a two-minute freedom for the interchange. The itinerary's timetables are identical for each weekday but different timetables are in force

at the weekend. Furthermore, traffic congestion and delays are usually more severe on weekdays, which could affect the probability of buses diverging from their schedules. Therefore, the data for the analysis have been captured during weekdays in March 2015 and have consisted of twenty records corresponding to twenty weekdays (Table V). Although there are twenty two weekdays in March 2015, the relevant data for the bus stop and the lines in question are missing on March 16th and 24th for some technical reason. These historical data become available after gathering raw bus data and pre-processing them according to the algorithm described in the previous chapter.

Table V. Weekday connections at the bus stop Tipotie from line 26 scheduled at 8:13 to line 27 scheduled at 8:16 on weekdays of March 2015.

Date	Arrival time of line 26	Departure time of line 27	Delta [a] (s)	Successful connection [b]
1	2	3	4	5
02.03.2015	08:15:30	08:16:40	70	+
03.03.2015	08:14:11	08:17:23	192	+
04.03.2015	08:14:42	08:17:39	177	+
05.03.2015	08:13:31	08:16:38	187	+
06.03.2015	08:14:38	08:15:39	61	+
09.03.2015	08:14:05	08:15:13	68	+
10.03.2015	*08:16:05*	*08:16:32*	*27*	-
11.03.2015	08:14:24	08:16:40	136	+
12.03.2015	08:15:01	08:17:05	124	+
13.03.2015	08:13:17	08:17:22	245	+
17.03.2015	08:14:50	08:19:29	279	+
18.03.2015	08:15:33	08:18:32	179	+
19.03.2015	08:14:19	08:15:54	95	+
20.03.2015	08:14:42	08:16:16	94	+
23.03.2015	08:14:46	08:17:19	153	+
25.03.2015	08:14:33	08:17:03	150	+
26.03.2015	08:12:48	08:16:21	213	+
27.03.2015	08:13:19	08:16:54	215	+
30.03.2015	08:14:52	08:17:06	134	+
31.03.2015	08:15:03	08:16:27	84	+

[a] Delta is a difference in seconds between departure time of the second line and arrival time of first line.

[b] Connection is successful if the difference between delta and the sixty-second limit for the change has a positive value.

After definition of the scope of observable data, we need to set the prior for the Bayesian inference. Having experience in taking buses in Tampere and considering findings of the studies made on bus data in Tampere [Kerminen et al., 2014; Syrjärinne et al., 2014; Syrjärinne et al., 2015] previously, we are aware of the probability that buses can fall behind or get ahead of their schedule in Tampere. On the other hand, this

itinerary has been suggested by the journey planner of Tampere and it has a sufficient interchange freedom of two minutes. Taking into consideration this information, let us assume that this connection is quite likely to succeed. For example, 80% probability of success can be a good guess.

The beta distribution characterized by two shape parameters α and β can be used as the conjugate prior distribution for binomial probabilities in Bayesian statistics [Kruschke, 2011; Owen, 2008]. When used in Bayesian analysis, α may be considered as the prior number of successes and β may be considered as the prior number of failures. Thus, expressing our belief in 80% successful connection as the parameters of binomial distribution, we say that we should have sixteen successful connections out of twenty cases within twenty weekdays of one month.

Based on the defined prior and the observed data for twenty weekdays in March on the itinerary in question, we evaluate the probability density function (Fig. 3). All in all, the Bayesian analysis requires several steps – assumption of the prior belief over possible probabilities of the successful connection, transformation of observed arrival times' and departure times' data to a binomial form, and inferring the posterior distribution of our beliefs using the Bayes' rule.

Fig. 3 shows the three steps of the analysis with the use of the equations (1)-(3). First, we define the parameters of the prior density function (1). Believing in 80% success of bus connection during one month or 20 weekdays, we set the parameters α and β equal to 16 and 4. It forms the beta distribution B(Θ|16, 4). Second, we define the likelihood function (2) using the observed data. In the experiment we have the number of successful connections z equal to 19, and the overall number of records in the observed data N equal to 20. Third, we define posterior density function (3). In the experiment the posterior function follows beta distribution B(Θ|35, 5). As a result of calculations, the expected probability of connection in this route is 87.5%, which is the posterior mean (4). The frequentist probability (5) of this connection is 95%.

$$P(\Theta) = B(\Theta \mid \alpha, \beta) \tag{1}$$

$$P(D|\Theta) = \Theta^{z}(1-\Theta)^{(N-z)} \tag{2}$$

$$P(\Theta \mid D) = B(\Theta \mid \mathrm{z} + \alpha, \mathrm{N} - \mathrm{z} + \beta) \tag{3}$$

$$P(D=1) = \frac{z+\alpha}{N+\alpha+\beta} \tag{4}$$

$$P(D=1) = \frac{z}{N} \tag{5}$$

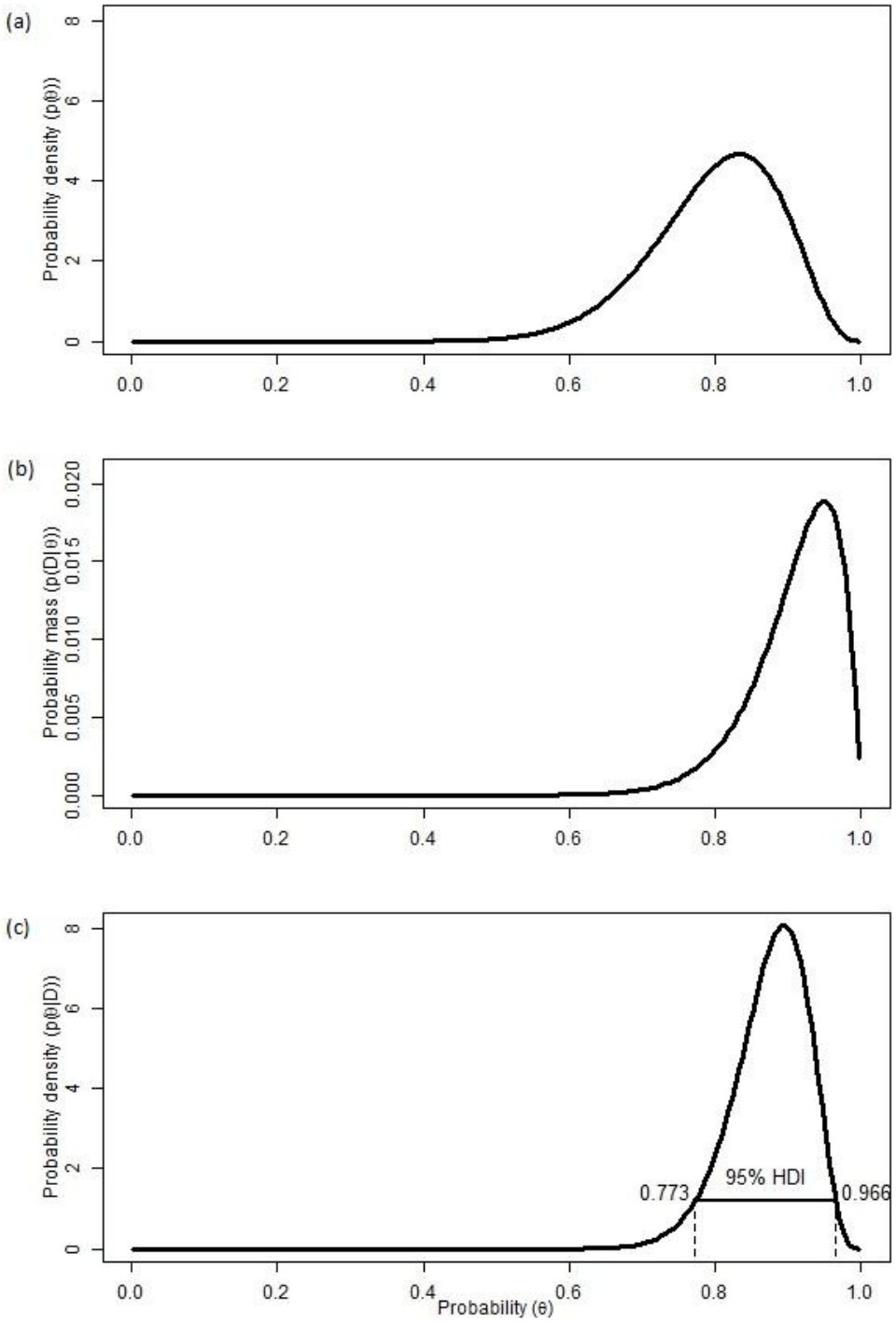

Figure 3. The prior (a), the likelihood (b) and the posterior (c) curves for the connection at Tipotie in March 2015

Bayesian analysis presents a higher stability of a probability of a variable when it is computed with a small amount of observed data. Without a doubt, a good guess about

the prior results in a better stability and coherence of a predicted value to the reality. Unlike the frequentist probability, which produces results close to reality only when computed with lots of data, Bayesian distribution should be able to update the prior and achieve a reasonable accuracy of prediction based on just a few new records. We also estimate the frequentist probability and the beta distribution on bus data captured during the first week of March in order to check this hypothesis and compare both approaches. The expected probability based on one-week data is 84%, which is closer to the previously computed probability (87.4%) than the frequentist probability (100%) to the previous value (95%). This evidences less dependence of the Bayesian probability on the data and the amount of data than of the frequentist probability.

In general, the investigated journey is relatively safe in terms of connection because the chance of a traveller catching the second bus is quite high. The fact that the posterior expected probability in our example is far less than the frequentist probability demonstrates a value of the Bayesian approach. The difference in resulting values reveals the benefit of the Bayesian analysis when a small quantity of data can help one anticipate the future situation. Thus, if we obtain less data than we analyzed in the example, the chance of getting the 100% frequentist probability increases because it is very likely to pick up randomly only a successful connection in the route in question. Nevertheless, the 100% probability of a successful connection does not describe the real situation with this route correctly, because we did face one case of failure during the month. Thus, should we believe that we can make this journey if we get the 100% frequentist probability? We guess we should not. In contrast to a single value of the frequentist probability, the posterior curve specifies the probability density function with the Highest Density Interval (HDI) that brings meaningful probability distribution for the risk estimation. Thus, the 95% HDI limiting the posterior curve in Fig. 3 defines the most probable success of the connection from 77% to 97% with 95%-confidence based on the data from March 2015. This continuous representation of the risk can bring added value for travellers who want to match the probable risk and their own willingness to take a risk in advance. While applying this framework, a researcher can choose between a single probability or continuous probability representations to report a forecast risk to users.

4.1.2. Validation factors

We can use different validation factors in order to measure how well the prior fits the data observed and how well the predicted value is in line with the real situation. Gelman et al. [1996] make the point that all models are generally wrong, but some

models can fit specific aspects of a problem reasonably well. A model validated by a number of factors will be preferred among others.

As the first validation factor, we assess the Bayes factor [Kass and Raftery, 1995]. The Bayes factor is a practical tool of applied statistics representing the weighted total probability of the new data across all possible parameter values weighted by its prior probability (6).

$$K = \frac{P(D \mid M1)}{P(D \mid M2)} \tag{6}$$

where P(D | M1) is the likelihood of the data D given the model M1, and P(D | M2) is the likelihood of the data D given the model M2. If K>1, then M1 is more strongly supported by the data than M2. If K>3, then the evidence in favour of M1 is substantial. If K>10, then the evidence in favour of M1 is very strong. Generally speaking, the greater the value K is, the more strongly the model M1 is supported by the data.

In our case, M1 is a model with a defined prior, while M2 is a model with an unknown prior. The likelihood of M1 can be calculated according to the binomial distribution (2) where probability relates to the prior distribution. Again, we need to make a guess about the prior distribution. We remember that we have set the parameters α and β equal to 16 and 4 yielding the mean of 0.8. The alternative hypothesis, or M2, relates to the belief that the probability of success is not equal to 80%. In order to calculate the Bayes factor, we have employed the *proportionBF* function of the "BayesFactor" R package [BayesFactor] suitable for binomial data. Thus, the Bayes factor for the data in our example with 19 successful connections out of 20 is equal to 1.91, which favours the null hypothesis about 80% prior belief although the Bayes factor value 1.91 is quite small to support this hypothesis strongly. In fact, smaller prior expectations about successful connections would produce a better model for the observed data.

The second validation factor of the model relates to the Posterior Predictive Assessment (PPA) [Kruschke, 2011, Gelman et al., 1996]. This method enables one to assess model fitness by measuring the discrepancy between the data and the posterior probability. Given a particular parameter value theta, we generate several new data sets using that value theta. Then, for each generated data set, we measure a discrepancy between the data and the parameter value theta. In our case, the parameter theta is the probability of successful connection. We simulate datasets with 20 connection attempts, and the discrepancy measure is the absolute difference between the expected amount of successful connections attempts and the amount of successful connections in the dataset. We measure the discrepancy for our observed data and compute the frequency, with which the generated-data-discrepancies equal or exceed the observed-data-discrepancy. This is called an estimate of the posterior predictive p-value.

Thus, we generated 10,000 datasets given the parameter value theta equal to random generation for the beta distribution with parameters corresponding to the posterior α and β. Then we calculated the posterior predictive p-value meaning that the amount of successful connections in the generated samples equals or exceeds 19 cases from 20 as in the observed data. As a result, the posterior predictive p-value equal to 0.33 is not large, however, it exceeds the significance level of 5%. In other words, the PPA test validated the current model. Fig.4 illustrates the frequency histogram of the test simulation. This output was highly expected since our prior belief does not differ considerably from the observed data. However, an overly pessimistic or, in contrast, overly optimistic belief in the success might cause far greater divergence between the posterior and the observed data, and hence, the frequency of the observed data can be too small in the simulation.

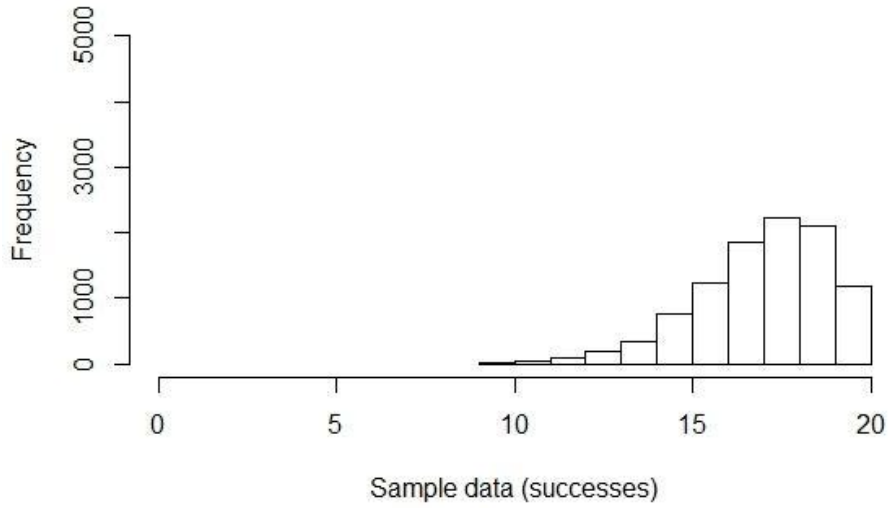

Figure 4. The PPA result for one-month data

All things considered, predictions on successful connections based on the prior belief in 80% of success have been validated by both the Bayes factor and the PPA characterizing the model positively. Although the values of both validation factors are quite modest, they are higher than the minimum values necessary for the null hypothesis acceptance. The process of validation can involve comparison of results obtained through the use of various input data and the prior belief. Thus, setting the prior correctly and varying the number of observations, we receive an opportunity to produce a better model. However, if validation factors have shown acceptable values in the first attempt, the model may be considered to be sufficiently good.

4.1.3. Case study

We have conducted a case study to test the proposed model and discover a minimal size of datasets for the analysis. Keeping in mind the goal to implement the model in a web or mobile application, we aim to minimize the quantity of data necessary for predictions. On the other hand, the data should be sufficient to make accurate predictions.

The case study involves picking up several itineraries with transfers, predicting the chance of connection for each itinerary, and validating the result. Table VI shows six bus routes selected for the case study. All the routes cross the centre of the city. All the connections take place during the 8 a.m. weekday rush hour in the areas close to the centre. This selection makes likely including connections with a different level of risk into the analysis.

We select four-week data from 02.03.2015 to 29.03.2015 and twelve-week data from 02.03.2015 to 24.05.2015 at the first stage of the analysis. The testing period comprises four days from 25.05.2015 to 28.05.2015 to estimate the posterior parameters at the second stage of the analysis. All in all, we analyze two datasets at the first stage (twelve and four weeks) and three datasets at the second stage (one day, two days, and four days) for each route. Although the number of weekdays in the week apart from holidays is constant, the number of available cases is not always equivalent to the number of weekdays in the chosen periods. The reason is that randomly missing signals from running buses due to technical issues with GPS-transmitters or Internet connections result in missing data.

Since bus stops can be located on different sides of the road or even in different streets, we have to consider a walking time between bus stops if a connection does not happen at the same bus stop. A walking time depends on the distance between bus stops, the need to cross a road or roads and to wait for a green pedestrian signal at the crossroad, and the average speed of a traveller. We agree with a default walking time proposed for each studied route by Tampere Public Transport Journey Planner [Tampere Journey Planner]. As a result, transfer time presents a sum of walking time between bus stops at the transfer and one-minute safety gap necessary to align a feeder vehicle and board a connecting vehicle. It means that the case study connections occurring at the same bus stop require one-minute transfer time, and ones involving different bus stops require a two-minute transfer.

The analysis includes twelve datasets at the first stage (Table VII) and thirty four datasets at the second stage (Table VIII). As we described above, the prerequisites to the analysis are selecting the routes with transfers, transforming a connection expressed as arrival and departure times to a binomial form, and defining the prior. These data enable us to estimate the posterior parameters, which can be stored and used subsequently as a

basis for the predictions based on the latest data as we have done at the second stage of the analysis. Thus, we can conduct the second stage using the posterior parameters α and β as the new prior. After we finish the Bayesian inference, we validate the models with the use of the Bayes factor and the PPA. The posterior predictive p-value serves as an estimate of the PPA test with the generation of 10,000 samples.

Table VI. Case study routes.

Parameters of connections	**Route numbers**					
	1	**2**	**3**	**4**	**5**	**6**
Origin	Epilä	Mediapolis	Tesoma	Pyynikki	Vihilahti	Varala
Destination	Keskisenkatu	Rauhanie mi	Valkama	TTY	Tieteenkatu	Virolainen
Feeder line	26	17	8	25	21	25
Connecting line	9	2	26	20	4	5
Change bus stop for a feeder vehicle	Keskustori H (0012)	Keskustori F (0010)	Tuulensuu F (0017)	Itsenäisyydenkatu 16 (0516)	Linja-autoasema (0522)	Keskustori F (0010)
Change bus stop for a connecting vehicle	Keskustori O (0007)	Keskustori F (0010)	Tuulensuu F (0017)	Itsenäisyydenkatu 16 (0516)	Linja-autoasema (0522)	Keskustori I (0042)
Scheduled arrival time of a feeder vehicle	08:20	08:33	08:15	08:28	08:13	08:42
Scheduled departure time of a connecting vehicle	08:23	08:35	08:18	08:31	08:18	08:50
Minimum transfer time (min)	2	1	1	1	1	2
Connection freedom[a] (min)	1	1	2	2	4	6
Transfer time[b] (min)	2	1	1	1	2	1

[a] Connection freedom is the difference between arrival time of a feeder vehicle, departure time of a connecting vehicle, and a transfer time.

[b] Transfer time is a one-minute gap plus walking time between bus stops in minutes.

Table VII. The first stage of the analysis.

Route number [a]	N weeks	Prior	N cases	N successes	Frequentist probability	Posterior mean	Posterior (α, β)	95% HDI	Bayes factor	p-value in the PPA
1	2	3	4	5	6	7	8	9	10	11
1	12	0.8	46	32	0.70	0.73	(48, 18)	(0.620; 0.831)	**1.20**	**0.70**
	4		14	10	0.71	0.76	(26, 8)	(0.623; 0.898s)	0.71	**0.75**
2	12	0.8	44	25	0.57	0.64	(41, 23)	(0.523; 0.755)	**63.53**	**0.81**
	4		18	9	0.50	0.66	(25, 13)	(0.508; 0.803)	**9.29**	**0.91**
3	12	0.8	49	46	0.94	0.90	(62, 7)	(0.827; 0.963)	**9.15**	0.34
	4		18	17	0.94	0.87	(33, 5)	(0.761; 0.964)	**1.56**	0.36
4	12	0.8	48	46	0.96	0.91	(62, 6)	(0.844; 0.972)	**24.61**	0.28
	4		17	16	0.94	0.86	(32, 5)	(0.755; 0.963)	**1.42**	0.35
5	12	0.8	48	33	0.69	0.72	(49, 19)	(0.614; 0.824)	**1.51**	**0.71**
	4		17	15	0.88	0.84	(31, 6)	(0.719; 0.946)	0.80	**0.50**
6	12	0.8	46	46	1.00	0.94	(62, 4)	(0.882; 0.988)	**424.21**	0.11
	4		17	17	1.00	0.89	(33, 4)	(0.793; 0.978)	**3.71**	0.20

[a] Route number refers to the header of Table VI.

The analysis reveals a few interesting findings. The model including only the first stage has already been validated for most of the cases. According to P-values in the PPA, in all cases the data does not deviate significantly from the predictions of the fitted model and, thus, the model has been able to fit the variation in the data. The values are largest for routes 1, 2, and 5, meaning that for those routes the model is able to predict success rates close to the observed data and/or variability large enough to contain the observed data. Interestingly, p-values are higher for smaller observable datasets in four routes out of six. In contrast, the Bayes factor is always higher and consequently more positive for larger datasets for all the routes. The Bayes factor obtained for four-week observed data of the first and fifth routes does not support our model. When it comes to the probabilities, as it was expected, the posterior mean and the frequentist probability are closer when we get larger datasets of observed data in all the cases. However, the differences between the posterior means calculated for a smaller and a larger dataset within one route and the same differences between the frequentist probabilities did not reveal any consistency although we hoped that the posterior mean would be more stable.

Table VIII. The second stage of the analysis.

Route number [a]	Prior[b]	N cases	N successes	Frequentist probability	Posterior mean	Posterior (α, β)	95% HDI	Bayes factor	p-value in the PPA
1	2	3	4	5	6	7	8	9	10
1	(48, 18)	4	4	1.00	0.74	(52, 18)	(0.640; 0.842)	**1.12**	0.32
		2	2	1.00	0.74	(50, 18)	(0.630; 0.837)	0.98	**0.53**
		1	1	1.00	0.73	(49, 18)	(0.625; 0.834)	0.96	**0.73**
	(26, 8)	4	4	1.00	0.79	(30, 8)	(0.660; 0.910)	**1.06**	0.41
		2	2	1.00	0.78	(28, 8)	(0.642; 0.904)	0.96	**0.61**
		1	1	1.00	0.77	(27, 8)	(0.633; 0.901)	0.96	**0.77**
2	(41, 23)	-	-	-	-	-	-	-	-
		2	0	0.00	0.62	(41, 25)	(0.505; 0.736)	**1.36**	**1.00**
		1	0	0.00	0.63	(41, 24)	(0.514; 0.746)	**1.05**	**1.00**
	(25, 13)	-	-	-	-	-	-	-	-
		2	0	0.00	0.63	(25, 15)	(0.476; 0.770)	**1.40**	**1.00**
		1	0	0.00	0.64	(25, 14)	(0.492; 0.786)	**1.06**	**1.00**
3	(62, 7)	4	4	1.00	0.90	(66, 7)	(0.836; 0.966)	0.95	**0.68**
		2	2	1.00	0.90	(64, 7)	(0.832; 0.964)	0.95	**0.81**
		1	1	1.00	0.90	(63, 7)	(0.829: 0.964)	0.97	**0.90**
	(33, 5)	4	4	1.00	0.88	(37, 5)	(0.783; 0.968)	0.96	**0.61**
		2	2	1.00	0.88	(35, 5)	(0.773; 0.966)	0.95	**0.78**
		1	1	1.00	0.87	(34, 5)	(0.767; 0.965)	0.97	**0.87**
4	(62, 6)	4	4	1.00	0.92	(66, 6)	(0.852; 0.974)	0.94	**0.71**
		2	2	1.00	0.91	(64, 6)	(0.848; 0.973)	0.95	**0.83**
		1	1	1.00	0.91	(63, 6)	(0.846; 0.973)	0.97	**0.92**
	(32, 5)	4	4	1.00	0.88	(36, 5)	(0.778; 0.967)	0.96	**0.61**
		2	2	1.00	0.87	(34, 5)	(0.767; 0.965)	0.95	**0.77**
		1	1	1.00	0.87	(33, 5)	(0.761; 0.964)	0.96	**0.87**
5	(49, 19)	4	4	1.00	0.74	(53, 19)	(0.634; 0.835)	**1.13**	0.30
		2	2	1.00	0.73	(51, 19)	(0.624; 0.829)	0.98	**0.53**
		1	1	1.00	0.72	(50, 19)	(0.619; 0.827)	0.96	**0.72**
	(31, 6)	4	4	1.00	0.85	(35, 6)	(0.746; 0.952)	0.98	**0.55**
		2	2	1.00	0.85	(33, 6)	(0.733; 0.949)	0.95	**0.72**
		1	1	1.00	0.84	(32, 6)	(0.726; 0.947)	0.96	**0.84**
6	(62, 4)	3	3	1.00	0.94	(65, 4)	(0.887; 0.989)	0.95	**0.84**
		2	2	1.00	0.94	(64, 4)	(0.885; 0.989)	0.96	**0.89**
		1	1	1.00	0.94	(63, 4)	(0.884; 0.989)	0.98	**0.94**
	(33, 4)	3	3	1.00	0.90	(36, 4)	(0.808; 0.980)	0.95	**0.73**
		2	2	1.00	0.90	(35, 4)	(0.803; 0.979)	0.95	**0.81**
		1	1	1.00	0.90	(34, 4)	(0.798; 0.979)	0.97	**0.90**

[a] Route number refers to the header of Table VI.
[b] Prior comes from the posterior mean calculated at the first stage.

The second stage of the analysis involved a variety of the prior parameters specific to each route since they were estimated before with the use of larger datasets. The aim of the second stage is to check whether it is legitimate to make predictions based on the

latest data covering from one to four days. The estimation revealed the fact that our approach based on Bayesian distribution requires more than four days of data to produce reliable results. Thus, the Bayes factor calculated for the majority of the datasets is smaller than one. It means that the specific connection probability comes a range of similar values. The PPA showed quite high p-values for all the cases, especially for one- and two-observed-day data. This may be because the inferred model allows enough variation around the estimated connection probability to contain the data, even if the probability does not match the data perfectly. Nevertheless, we must consider both validation factors when estimating the ability of the model to predict the future values sufficiently well.

Apart from the validation process, let us review the probabilities calculated with the different approaches and their dynamic when the size of the testing datasets becomes smaller. At the second stage of the analysis in this case study, we have N or zero successes from N cases due to very small datasets, which consequently leads to the extreme frequentist probabilities. Thus, the frequentist probability for all the routes except the second one is equal to 100%, and for the second route it is 0% (Table VIII). We cannot consider this result as credible because the figures available from larger datasets at the first stage provide the 100% frequentist probability only for the sixth route, and no route has the 0% frequentist probability given one- or three-month data. However, the expected probability in the Bayesian inference varies from the minimum 2% for the second route to the maximum 10% for the fifth route. The 2-10% variability of the connection risk is not enormous and therefore might still represent useful information for the user. The same cannot be said about the 0-100% variability, which we obtained with the frequentist approach. The dynamic of the expected probability within one route based on Bayesian inference is very low in the case study, thereby establishing the stability of the Bayesian probabilities regardless the size of observed data.

In conclusion, we provide some recommendations about applications of the proposed framework. If delays due to traffic congestion in the city are relatively light, the analysis can start with the 80% prior and capturing one-month bus movements' data. If the result of the validation process does not seem satisfactory, the iterative process of the prior and the size of dataset changes should be initiated. After a good prior and size of data for estimation of the posterior are discovered, we can store the distribution parameters in order to use them for the risk re-estimation daily or weekly based on the latest data. However, when it comes to the amount of the latest data sufficient for reliable predictions, we must take more than one-week data, and preferably, as much data as can be processed quickly for online services. The validation process that we have performed in the case study indicates that even four-day data is not enough. This

sensitivity of the framework to the size of data is highly undesirable for real-time applications and services, since the size of data in the analysis cannot be minimized safely.

Besides this, as long as we need to store the distribution parameters, we need to know all possible transfers in the city in advance at the initial stage of the analysis. This requirement can be hard to fulfil because a general approach of trip planners is to plan itineraries based on a graph pre-built with road map, bus routing, and scheduling static data. Thus, trip planners generate itineraries and consequently transfers in real-time but do not store transfer information anywhere. If there is a way to generate all transfers, the framework is recommended to be implemented in online journey planners. The risk connection prediction feature is original and able to move trip planning services to a new quality level when feasibility of a bus route is evaluated on actual data rather than static timetables or user's experience.

4.2. Cumulative distribution function

As a consequence of the previously discussed methods having serious limitations for implementation in trip planner applications, we have developed an alternative approach at estimates a connection chance based on parameters of two distributions rather than one. If the beta distribution function requires data in a binomial form, other standard distributions like normal or log-normal distributions deal with nominal data. Thus, if we know the distribution type of the bus arrival times and bus departure times in the city of Tampere, we can estimate a chance of a bus connection relying only on the distribution parameters of arrival times and departure times. We assume that distributions of arrival time of a feeder bus and departure time of a connecting bus are uncorrelated.

Without a doubt, bus delays, arrival times, departure times, dwell times, and travel times data within the same geographical location might follow different types of distribution. The difference in the data characteristics refers to the local conditions of the transportation network. Numerous external factors such as different congestion levels on road segments, number of boarding passengers, rush hours, day of the week, and weather conditions might impact on the transportation network performance and, consequently, on adherence of buses to their timetables.

The task of discovering the distribution type of data can be quite difficult. First, a researcher needs a lot of data available to execute functions of checking the fitness of data to standard distributions. These functions frequently have requirements on the minimum number of values in the input dataset, which can start from 100 and even more. Second, a researcher should design the experiment appropriately to separate data

to meaningful clusters. For example, if we deal with bus arrival and departure times, we need to study the data separately for different time and space coordinates. Thus, buses tend to be delayed more in the central areas of the city and during rush hours. On the other hand, there can be few bus stops in the route that are operating as the control points where a driver stops to align with the timetable if a bus runs ahead of the timetable. Besides this, the frequencies of delays and timely arrivals might differ on weekdays and weekends because road network's load and passengers plans are naturally different on weekdays and weekends. Ideally, each bus stop and each trip identifier should form its own data cluster when the data distribution is checked. In this case we can face a situation when data in different clusters fit better with different distribution types.

The problem of distribution type identifications can be clearly seen in Fig. 3 illustrating density curves of arrival times and departure times of specific vehicles at the bus stop. The shapes of the curves are not normal. Apparently, these curves do not resemble any standard distribution curve at all.

Nevertheless, if we cannot say that the observable data always follow some specific distribution, we can resort to approximations. Syrjärinne et al. [2015] discovered that, although bus arrival times in the city of Tampere do not strictly follow any standard distribution, both normal distribution and log-normal distribution approximated datasets sufficiently well. This finding allows us to treat bus arrival times in Tampere as normally distributed. The common logic also leads us to an assumption that departure times should have more or less the same nature as arrival times since the time that buses spend within the area of the bus stops is usually minimal. There are few exceptions, so called control points, where buses have to wait if they run ahead of the schedule. The normal distribution assumption is more beneficial than log-normal or another distribution since the data do not require any transformation, and normal distribution parameters, which are mean and variance, are easy to compute.

In order to estimate the probability of a connection, we operate with arrival times of a feeder bus at the alighting stop and departure times of a connecting bus from the boarding bus stop. Let us assume that each trip identified by a specific line, origin, destination, and origin aimed departure time has unique distribution parameters for each bus stop in its route. When we split the data to such clusters, we are able to analyze separately all the location and time coordinates in the area of our interest. Hence, knowing the parameters of distribution of arrival and departure times for each trip identifier and each bus stop code, and the scheduled transfer time for the connection bus stops, we can estimate the probability of a connection. We suppose that a connection will be successful if the difference between the distribution of departure times from the

boarding stop of a connecting vehicle D_2 and the distribution of arrival times at the alighting stop of a feeder vehicle D_1 is larger than the scheduled transfer time T (7).

$$D_2 - D_1 > T \qquad (7)$$

Then the probability of a successful connection or the inequality (7) can be expressed through the cumulative distribution function. If M_1 and V_1 are mean and variance of distribution D_1, M_2 and V_2 are mean and variance of distribution D_2, then we can transform the inequality (7) to the equation (8), which defines the probability of a successful connection.

$$P(D_2 - D_1 > T) = 1 - pnorm(T, M_2 - M_1, V_1 + V_2) \quad (8)$$

In the equation (8) *pnorm* is the cumulative normal distribution function. *Pnorm* function evaluates the tail area of the standardized normal curve from minus infinity to T. We apply the algorithm of *pnorm* estimation proposed by Hill [1973].

In other words, having processed data beforehand and stored only M and V parameters for each trip identifier and each bus stop, we can calculate the distribution function in real time when users send requests to the application. The heavy computations related to the data cleaning and pre-processing discussed in chapter 3 are still necessary, but the actual estimation of a connection chance becomes easy and possible to be done in real time. Such an approach gives us an opportunity to skip intensive real-time calculations and transfer the heaviest burden of big data processing from an online journey planner, which is sensitive to data size, to a separate processing module which can handle a large amount of data. All things considered, this method overcomes the limitation of the Bayesian beta distribution, but it should still achieve accurate results provided the statistical properties of the data being studied properly and expressed correctly in the cumulative distribution function.

4.3. Comparison of the methods

An important question is how we can compare the accuracy of all the methods discussed above. Whereas the Bayesian analysis provides a number of validation factors of the models, the frequentist probability and cumulative distribution function estimate the frequency and difference between probability distributions, which are accurate by design. The basis for the comparison will refer to finding the range or difference between connection chances predicted for the same journey and bus stop by each method. Thus, we need to estimate the probabilities obtained from each method and the same datasets. We also have to assess the posterior mean as a resulting value in Bayesian inference since the other two methods present the predicted variable in a nominal single value rather than a range.

We prefer to use the same routes that we have investigated in the case study in section 4.1.3 for the ease of the case study description. Taking the same routes as a base of the methods' comparison allows us to engage already prepared and studied data and to describe the case study briefly since the data necessary for the analysis have already been described. Furthermore, we have already predicted the probabilities of these connections calculated with the use of the datasets of different sizes and periods, therefore any extreme values obtained in the current case study can be compared with the previous case study.

The input parameters for the functions are as follows. We select 60-day observable data for predicting a chance of connection by each method. In fact, there will be far less cases due to various timetables on Saturdays, Sundays, and missing data. As long as we do not know beforehand how many cases we can obtain from the observable data, we need to select a number of days that we will scan in order to find the necessary routes and connections. For the Bayesian analysis, we set the prior to 80% (check section 4.1.1 for details). For the cumulative distribution function we set minimum transfer time defined for each connection in Table VI. The data are to be cleaned and pre-processed according to the algorithm explained in chapter 3 before we can use them at the processing step.

Table XI presents the results of calculations. Fig 5. makes the differences between chances of connections predicted by the three different methods visible. As we can observe, the differences between predicted values are not large in most cases. The normal distributions difference is closer to the frequentist probability for the first, second, third and sixth routes. However, it is closer to the posterior mean for the fifth route. We can also see the outstandingly odd prediction made by the cumulative distribution function for the fourth route. Having 34 successful connections from 36 cases in the observable dataset, we obtain only 81% of a successful connection chance predicted by the cumulative distribution function, whereas the first two methods are apparently more optimistic. The reason for such a big difference might lay in the relatively large standard deviations in comparison to the sum of means of arrival and departure times for this route.

All in all, as far as the predicted values for most cases do not differ drastically, we can conclude that all the methods have the capability to predict the chance of a bus connection. The question is what method is better for implementing in a mobile or web trip planner sensitive to the size of data and complexity of calculations. As we discussed in the previous chapters, the first two methods require us to know in advance all the possible connections that can be proposed by a trip planner. It is a strict limitation for the application development. If we do not store the information about the successful connections in our historical datasets, we will have to process large batches of historical

data to predict the chance of connection for each user's request when the actual trip planning is happening. Here we face a dilemma as to whether big data should or should not be processed in real time. In our opinion, a real time application should not handle such workloads because application performance and response time are extremely crucial for end users. Having numerous fast-responding trip planners at their disposal, users will not be willing to use a new slow trip planner. Thus, our application will not be able to compete with existing trip planners regardless of whether it has original and useful predictive functionality. All things considered, we believe that the cumulative distribution function is the best-among-the-discussed methods that can be integrated into a trip planner to enable it to predict a chance of a bus connection without a loss of performance.

Table XI. Comparison of connection chances predicted by the three methods based on 60-day data from March-April 2015.

Route number [a]	**Observed data**		**Predicted chance of connection (%)**		
	Number of cases	**Number of successful connections**	**The frequentist probability**	**The posterior mean**	**The normal distributions difference**
1	2	3	4	5	6
1	33	23	70	74	69
2	31	17	56	65	53
3	36	34	94	89	94
4	36	34	94	89	81
5	36	23	64	70	70
6	33	33	100	92	100

[a] Route number refers to the header of Table VI.

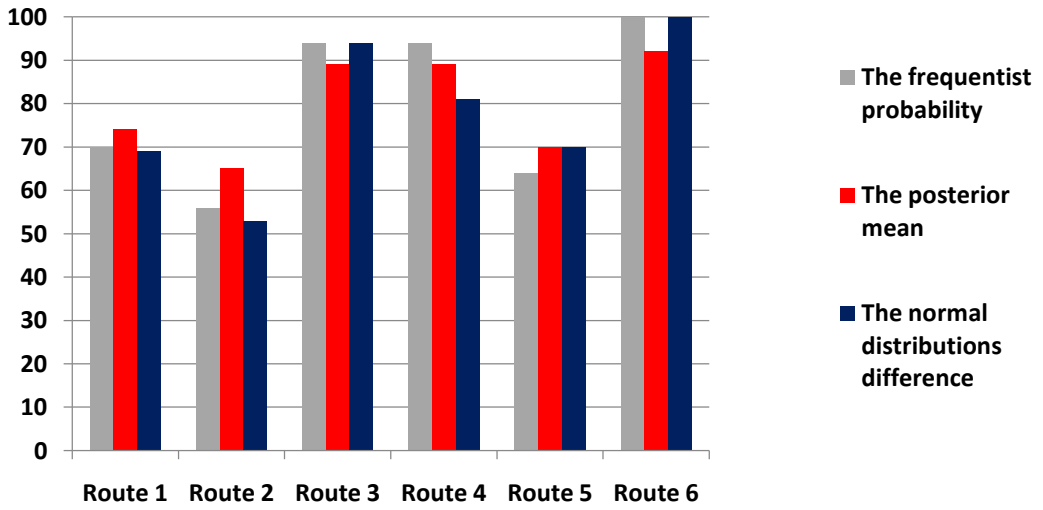

Figure 5. The probabilities of successful bus connections.

4.4. Implementation of bus connection risk estimation in the trip planner

The probability of a bus connection estimated as a cumulative distribution function seems to be the best option among the methods discussed in the previous sections. Therefore, we selected this method for implementation of the web trip planner in order to visualize the predicting capability of a bus connection risk.

We integrated the method of risk estimation based on the cumulative distribution function and the assumption that arrival times and departure times of buses in the city of Tampere follow the normal distribution into the open source trip planner [Open Trip Planner]. Open Trip Planner is an open source platform for multi-modal and multi-agency trip planning. It relies on open data in open standard file formats, making it easy to deploy in any city for which the GTFS and Open Street Map data are available. Besides a REST API for journey planning, it includes a map-based Java script client, which we deployed in the web server with a port number 65303 [Tampere Open Trip Planner]. Open Trip Planner creates travel time contour visualizations based on input parameters about the origin, destination, travel mode, and time. This trip planner is relatively simple to deploy in order to get basic trip planner's functionality for a chosen geographical area in the web application. It also offers a good support for developers and a set of technical documentation with version control.

The stable version 0.11 of Open Trip Planner served as a base for our trip planner. The planning graph has been built based on the latest GTFS files downloaded from the file repository of ITS Factory [GTFS for Tampere] and the OSM file for the region of Tampere [Open Street Map]. In fact, the graph is rebuilt and updated with the latest distribution data on a daily basis, although new GTFS files are integrated into the graph in general twice a year when the bus timetables change.

In addition to Open Trip Planner, we employed MapReduce framework, which is described in detail in chapter 3, for calculating the distribution parameters of arrival times and departure times for every unique journey and bus stop in the route of the journey. MapReduce environment is chosen for two reasons. First, the task of calculating distribution parameters requires a lot of data to be processed. Second, this task should be repeated automatically on a daily basis with a minimum workload of a web server where the trip planner application runs. MapReduce can efficiently handle big data to aggregate them to quite small datasets which can be added to the trip planner's graph without performance loss. Because the response time in web applications is critical in keeping users loyal, real-time processing of big data is not a good choice. The reasonable solution is to process all data in advance in a different system and transfer only the summarized data to a web application. If necessary, this summarized result can be updated from time to time with the latest real-time data.

Our program processes daily the latest sixty files with arrival and departure times, which are equal to sixty-day data, in order to calculate the normal distribution parameters for each trip identifier and each bus stop. Data processing is done regularly by the MapReduce task described in Fig. 6. The running time of the task takes up to two minutes, less at the weekend or during holidays. As a result, the data of around 300 Mb are converted to one file of around 5Mb. The number of days from which data are selected as input of the MapReduce task might vary according to the local conditions. Thus, if the seasons in the area are clearly distinct, the weather conditions impact frequently on traffic fluency. Then it makes sense to adjust the period of data needed for the analysis to the time of more or less similar weather conditions. Consequently, the period of data can also vary during the year. We figured out experimentally that one or two months of historical data were sufficient to make credible bus connection predictions in the Tampere region.

The parameters of our MapReduce program can be easily adjusted through the configuration file. This file contains information about the paths to files in the local machine as well as in HDFS. Additionally, the number of days for estimating distribution parameters and the starting date to form the arrival times' file can be changed. If necessary, new parameters such as data distribution type may be added to the configuration file with minimum correction of the algorithm, making deployment of the proposed system possible and rather easy in a new cluster and new geographical area. We plan to distribute the code openly.

The only issue that has not been fully resolved in our trip planner concerns the relationships between different sources of data described in chapter 3. As long as GTFS files serve as a base for building the planning graph, each unique trip in the graph receives a separate 10-digit identifier. However, Journeys API does not contain this identifier. That is why the proposed framework includes an extra stage in order to integrate GPS-based bus data distribution parameters into the graph. At this stage, our algorithm searches for a trip identifier for each unique set of line, direction, origin stop, destination stop, and origin aimed departure time in the specific GTFS files. Once discovered, a trip identifier is added to the original file with the distribution parameters. The insurmountable problem of this operation is that Journeys API does not provide letters for a few lines, which are identified as a compound of a number and letter in GTFS data. Thus, sometimes the trip planner is unable to find distribution parameters for a journey. In this case, the trip planner outputs an "undetermined" chance of connection for an itinerary.

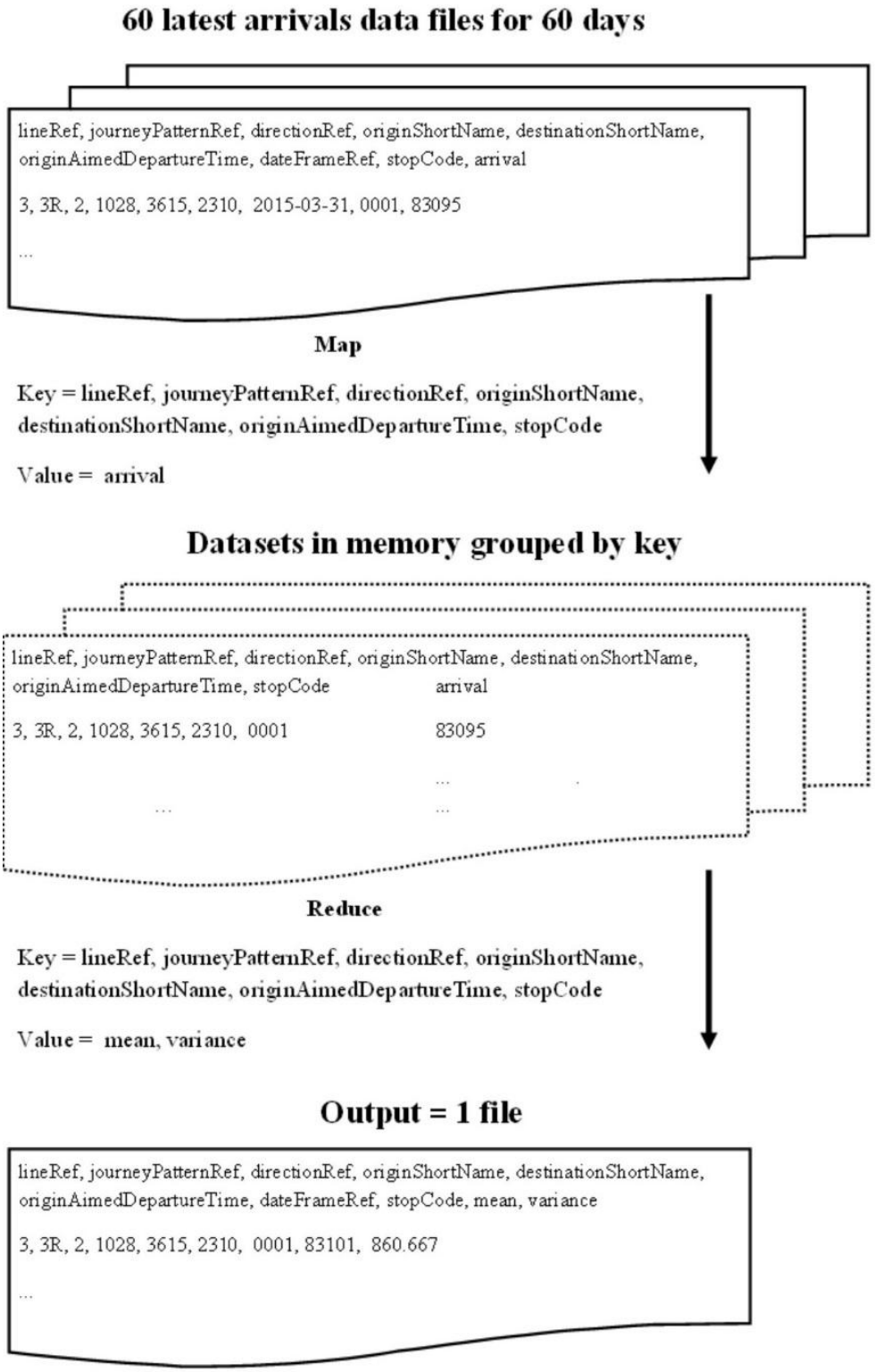

Figure 6. MapReduce task for getting distribution parameters data.

We changed several classes and functions in the original code of Open Trip Planner in order to add risk estimation functionality. The majority of changes affected core modules of graph loading and trip planning. The front-end Java script modules underwent minor changes to enable visualizing new information for the trip summary section in the “Itineraries” widget. One example of the framework in action is illustrated

in Fig. 7, which is to be interpreted as follows. If an itinerary consists of one or more transfers, the cumulative distribution function (8) estimates the probability of a bus connection or connections with the minimum transfer time depending on the walking distance between transfer bus stops. The current version of Tampere Open Trip Planner calculates the difference between distribution of arrivals times, but a coming version will use the distribution of departure times as described in section 4.2 and the equation (8). The new function of connection chance estimation is original in trip planning since currently no existing trip planner provides such information yet.

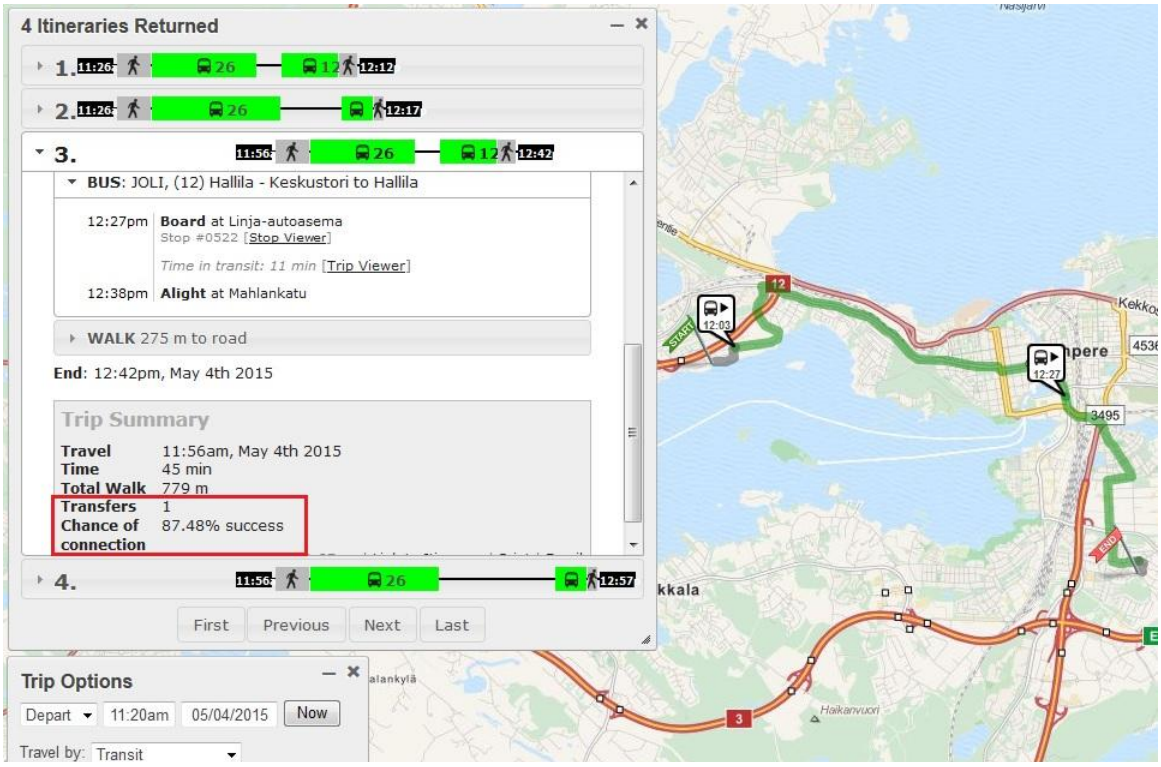

Figure 7. An example of a bus connection chance estimation in the application.

The response time for a user's request does not show a great difference in comparison to the response time before the trip planner's code changing. The response time rose by a maximum of 0.1 s. All in all, the proposed framework for bus connection risk estimation influences quite little on overall performance and usability of the application. We consider the connection risk prediction feature as highly beneficial for end users of the trip planner: the new information brings value to end users while changes do not have any drawbacks.

The general idea is to make the whole system work automatically. There are several steps of execution (Table X) necessary for data gathering, cleaning, pre-processing, and processing, and finally feeding the trip planner with the relevant summarized data. After all the steps are done, the trip planner becomes capable of consulting end users about a bus connection risk. The first two steps relate to gathering date on real-time bus

movements via Journeys API and copying these data as hourly CSV files to the HDFS of the distributed cluster. The third step includes data cleaning and pre-processing. Thus, the previous day's raw data are processed to form a CSV file with arrival times and departure times for each unique journey every night at the least intense time of the cluster and web server. After this step, the distribution parameters are recalculated to consider the new file with the pre-processed data. The fifth step refers to finding matches between GTFS-based trip identifiers and GPS-based pre-processed records when the new file with distribution parameters and graph-compatible trip identifiers is formed. Last but not least, the trip planner graph is reloaded into the web server memory with the use of freshly calculated distribution parameters.

Table X. Updating trip planner graph's steps.

#	Task	Regularity	Number of input files	Average input data size (Mb)	Average output data size (Mb)	Average running time (min)
1	2	3	4	5	6	7
1	Capturing real-time bus movements data from Journeys API	once per second	-	-	-	-
2	Copying real-time bus movements data to HDFS	once per hour	-	-	27	0.05
3	Estimating arrival times for each trip and each bus stop	once per day	24	650	5	7
4	Calculating distribution parameters of arrival times data	once per day	60	300	5	2
5	Finding a trip identifier in GTFS files for each trip	once per day	1	5	19	2
6	Updating the trip planner graph with the new distribution parameters	once per day	1	19	-	2

To sum up, the aim of making the system work automatically can be easily achieved if one executes the described steps as jobs on the cluster and web server. The only manual work necessary for the proper operation relates to copying new GTFS files to the cluster repository when the bus schedule changes. The calendar file of GTFS and changes of routes and timetables create the need to keep the planning graph nodes up-

to-date any time. Otherwise, the graph will not contain new dates, routes, or timetables to be able to generate relevant itineraries. Bus timetables of Tampere usually change twice a year to switch between a regular schedule and a special summer schedule when fewer buses are running in the routes. At the time of writing this thesis, Tampere Open Trip Planner's graph used the autumn 2015's GTFS files.

5. Conclusions and discussion

Estimation of bus connection risk is novel and original in the transportation field. So far there is no existing trip planner providing predictions on a bus connection. However, as shown in this thesis, it is possible to create such a trip planner with the use of open data. Following the framework described in the present study, one can build his or her own trip planner application provided the map and bus data are open for the city of interest. Thus, bus connection chance estimation integrated into a trip planner will make travellers better informed and offer them a tool for a more intelligent use of a transportation network.

This thesis has reviewed different methods and models widely used in transportation for predicting future values of different public transportation variables. In our opinion, the best classification of the models should be based on the nature of data and processing algorithms. Thus, static models engage historical static datasets and algorithms that can handle large and unchanging data. In contrast, dynamic models rely on real-time small data updates and algorithms, whose testing phase takes far less time than a training phase. In this way, an algorithm builds a model with the use of large historical datasets beforehand. Once constructed, a model is updated with small portions of real-time data, which enables it to correct the model behavior depending on the changes of observable variables.

Although the accuracy of dynamic models seems in principle to be higher than that of static models, previous studies have illustrated that static models can outperform dynamic models in terms of accuracy. Furthermore, if talking about connectivity risk, users are usually interested in the connection risk of a planned journey before they start travelling. For example, when it is highly crucial to arrive at the destination in time, planning should take place in advance in order to compare different alternative routes and choose the most reliable one. Thus, if a user is planning far before the actual trip starts, the real-time situation in the transportation network is not very valuable for the prediction. All things considered, static models produce sufficiently good predictions of a bus connection chance.

The majority of models describes or predicts arrival times, travel times, and dwell times, whereas the probability of a bus connection is not so well-studied. This thesis has attempted to fill the gap between the knowledge of the connectivity risk and the need for bus connection estimations in end-users' applications. In order to do this, this study has elaborated two original frameworks. The first framework is based on the Bayesian analysis. The second framework incorporates the cumulative distribution function. Having compared the frequentist probability, Bayesian inferring, and the cumulative distribution function, this study has discovered that all three methods produced quite

close predictions for the routes of the case study. Without a doubt, it is very difficult to conclude which method provides the most accurate result. A positive sign is that the difference between the values predicted by all the three methods is not enormous. The predicted chances of bus connections stay close and seem sensible for most routes. Perhaps, future work can focus on some better measures of how different approaches can be compared. The testing in this thesis concentrates in how well the models describe the data. With enough data and few parameters, this may be close to testing for how well the method generalizes. Proper testing for how well the methods described here generalize is not a minor task and it is left out of this thesis.

In general, the frequentist probability is not stable when the datasets are small. Besides being sensitive to the size of observed data, both the frequentist probability and Bayesian analysis methods require knowledge of all possible transfers in advance in order to be able to save the parameters of the transfers for real-time users' requests in a trip planner. The framework based on the cumulative distribution function does not need this information since the prediction can be made in real time requests as the difference between arrival times distribution and departure times distribution. In other words, the heavy calculations of distribution parameters can be fulfilled in a separate system, and only a small set of distribution parameters for each journey and each bus stop has to be loaded to a web server's memory where the trip planner runs.

Our recommendation is to use the bus connection risk estimation framework based on the cumulative distribution function as the best one for implementation in applications. As a part of the thesis work, we deployed an open source trip planner for the city of Tampere and integrated the function of connection chance estimation for visualization purposes. Arrival time and departure time data in the city of Tampere are considered normally distributed. Therefore, the program calculates the normal distribution parameters of bus arrivals and departures as the key components of the predictive method.

The work on deploying the system comprises several steps. The first step is to gather GPS-originated data on bus movements in real time by polling Journeys API once per second. Then the MapReduce program should process raw data in the distributed cluster. Data pre-processing includes cleaning the data, finding arrival and departure times and discarding unnecessary data. Then the second MapReduce program calculates the distribution parameters. The parameters of the program can be easily adjusted to one's needs through the configuration file. Then the newly calculated parameters should be loaded into the planning graph of the trip planner. The idea is to reload the graph every night in order to update the distribution parameters based on the latest sixty-day bus data. The whole system can work automatically due to the scheduled jobs unless a new bus schedule is issued. The overall running time for

calculating parameters and loading them to the web application is thirteen minutes on average. Downtime of the trip planner does not exceed two minutes. In any case, it can be done during the quiet hours of the night; therefore the downtime of the application should not be critical for end users.

All in all, an efficient framework proposed in this thesis opens a new horizon for trip planning applications. The application developed as a part of this thesis has received a national level award in the Open Finland Challenge 2015 competition, which took place in Helsinki. Our trip planner with estimation of bus connection chance has been recognized as a novel and original solution for the future development of ITS. So far, web and mobile trip planners have focused on journey planning linked to static timetables. The predictions in existing applications have covered bus arrival times but not the probabilities of connections. However, even such predictions are possible provided that relevant bus data are available. Otherwise, only the user's experience about specific bus lines and time can help guess if the trip can be successful. Trip planners with connection risk estimation will enable consulting travellers about trip feasibility and to increase their loyalty to public transport in general. The recommended framework consists of a theoretical description and practical application which makes the study very useful not only for researchers, but also for transportation system decision-makers, software developers, and bus users. The framework can be utilized to build the same service by anyone in any city where map and bus data are open.

References

[Abdelfattah and Khan, 1998] Ali Abdelfattah and Ata Khan, Models for predicting bus delays, *Transportation Research Record*, vol. 1623 (1), 8-15.

[Alves et al., 2012] David Alves, Lius M. Martinez, and Jose M. Viegas, Retrieving real-time information to users in public transport networks: an application to the Lisbon bus system, *Procedia – Social and Behavioral Sciences*, vol. 54, 2012, 470-482.

[Baptista et al., 2011] Arthur Trigueiro Baptista, Eric Bouillet, Francesco Calabrese, and Olivier Verscheure, Towards building an uncertainty-aware personal journey planner, in *Conf. Rec.2011 14th International IEEE Conference on Intelligent Transportation Systems*, 2011, 378-383.

[Batley and Ibanez, 2012] Richard Batley and J. Nicolas Ibanez, Randomness in preference orderings, outcomes and attribute tastes: An application to journey time risk, *The Journal of Choice Modelling*, v. 5, 2012, 157-175.

[BayesFactor] BayesFactor, An R package for Bayesian data analysis URL http://bayesfactorpcl.r-forge.r-project.org/ [Accessed on April 2016]

[Bian et al., 2015] Bomin Bian, Ning Zhu, Shuai Ling, and Shoufeng Ma, Bus service time estimation model for a curbside bus stop, *Transportation Research Part C: Emerging Technologies*, vol. 57, 2015, 103-121.

[Borole et al., 2013] Nilesh Borole, Dillip Rout, Nidhi Goel, Dr. P. Vedagiri, and Dr. Tom V. Mathew, Multimodal public transit trip planner with real-time transit data, *Procedia – Social and Behavioral Sciences*, vol. 104, 2013, 775-084.

[Ceder, 2007] Avishai Ceder, *Public Transit Planning and Operation: Theory, Modeling and Practice*. Burlington, MA: Elsevier, 2007, 365-399.

[Chandra and Quadrifoglio, 2013] Shailesh Chandra and Luca Quadrifoglio, A model for estimating the optimal cycle length of demand responsive feeder transit services, *Transportation Research Part B: Methodological*, vol. 51, 2013, 1–16.

[Chen et al., 2013] Dewang Chen, Long Chen, and Jing Liu, Road link traffic speed pattern mining in probe vehicle data via soft computing techniques, *Applied Soft Computing*, vol. 13 (9), 2013, 3894-3902.

[Chien et al., 2002] Steven I-Jy Chien, Yuqing Ding, and Chienhung Wei, Dynamic bus arrival time prediction with Artificial Neural Networks, *Journal of Transportation Engineering*, vol. 128 (5), 2002, 429-438.

[Fu et al., 2014] Xiao Fu, William H.K. Lam and Bi Yu Chen, A reliability-based traffic assignment model for multi-modal transport network under demand uncertainty, *Journal of Advanced Transportation*, vol. 48, 2014, 66-85.

[Gelman et al., 1996] Andrew Gelman, Xiao-Li Meng, and Hal Stern, Posterior Predictive Assessment of model fitness via realized discrepancies, *Statistica Sinica*, vol. 6, 1996, 733-807.

[Grotenhuis et al., 2007] Jan-Willem Grotenhuis, Bart W. Wiegmans, and Piet Rietveld, The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings, *Transport Policy*, vol.14, 2007, 27-38.

[GTFS for Tampere] GTFS files from ITS Factory. URL http://data.itsfactory.fi/files/gtfs/ [Accessed on August 2015]

[Guardiola et al., 2014] Ivan G. Guardiola, Teresa Leon, and Fermin Mallor, A functional approach to monitor and recognize patterns of daily traffic profiles, *Transportation Research Part B: Methodological*, vol. 65, 2014, 119-136.

[Hans et al., 2015] Etienne Hans, Nicolas Chiabaut, Ludovic Leclercq, and Robert L. Bertini, Real-time bus route state forecasting using particle filter: an empirical data application, *Transportation Research Procedia*, vol. 6, 2015, 434-447.

[Hill, 1973] I. D. Hill, Algorithm AS 66: The normal integral, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 22 (3), 1973, 424-427.

[Hunter at al., 2009] Timothy Hunter, Ryan Herring, Pieter Abbeel, and Alexandre Bayen, Path and travel time inference from GPS probe vehicle data, *NIPS Analyzing Networks and Learning with Graphs*, vol. 12, 2009.

[ITS action plan and directive, 2010] ITS action plan and directive, Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport, European Commission.

[ITS Leeds, 2008] ITS Leeds, Imperial College, John Bates Services, 2008. Multimodal Travel Time Variability. Final Report to the Department for Transport.

[Kass and Raftery, 1995] Robert E. Kass and Adrian E. Raftery, Bayes Factors, *Journal of the American Statistical Association*, vol. 90 (430), 1995, 773-795.

[Kerminen et al., 2014] Riitta Kerminen, Esa Hakulinen, Paula Syrjärinne, Jyrki Nummenmaa, Peter Thanisch, and Ari Visa, Analysis of bus delays in Tampere using real-time data, in *Proc. 10thITS European Congress*, Helsinki, 2014.

[Kim and Mahmassani, 2015] Jiwon Kim and Hani S. Mahmassani, Spatial and temporal characterization of travel patterns in a traffic network using

vehicle trajectories, *Transportation Research Procedia*, vol. 9, 2015, 164-184.

[Kim and Schonfeld, 2014] Myungseob Kim and Paul Schonfeld, Integration of conventional and flexible bus services with timed transfers, *Transportation Research Part B*, vol. 68, 2014, 76-97.

[Kruschke, 2011] John K. Kruschke, *Doing Bayesian Data Analysis*. Elsevier Academic Press, 2011, 77–90.

[Kumar et al., 2013] B. Anil Kumar, Lelitha Vanjakshi, and Shankar C. Subramanian, Day-wise travel time pattern analysis under heterogeneous traffic conditions, *Procedia – Social and Behavioral Sciences*, vol. 104, 2013, 746-754.

[Lian and Chen, 2013] Xiang Lian and Lei Chen, Trip planner over probabilistic time-dependent road networks, *IEEE Transactions on Knowledge and Data Engineering*, vol. 6 (1), 2013, 1-14.

[Lo et al., 2006] Hong K. Lo, X.W. Luo, and Barbara W.Y. Siu, Degradable transport network: Travel time budget of travellers with heterogeneous risk aversion, *Transportation Research Part B: Methodological*, vol. 40, 2006, 792-806.

[Mazloumi et al., 2011] Ehsan Mazloumi, Geoff Rose, Graham Currie, and Majid Sarvi, An integrated framework to predict bus travel time and its variability using traffic flow data, *Journal of Intelligent Transportation Systems*, vol. 15 (2), 2011, 75-90.

[Muller and Furth, 2009] Theo H.J. Muller and Peter G. Furth, Transfer scheduling and control to reduce passenger waiting time, *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2112, 2009, 111–118.

[Ng et al., 2011] Man Wo Ng, W.Y. Szeto, and S. Travis Waller, Distribution-free travel time reliability assessment with probability inequalities, *Transportation Research Part B: Methodological*, vol. 45, 2011, 852-866.

[Open Street Map] City-sized portions of Open Street Map data. URL https://mapzen.com/data/metro-extracts [Accessed on February 2015]

[Open Trip Planner] Open Trip Planner. URL http://www.opentripplanner.org/otp/ [Accessed on February 2015]

[Owen, 2008] Claire B. Owen, *Parameter estimation for the beta distribution*, M.Sc. thesis, Brigham Young University, December 2008, p. 49.

[Patnaik et al., 2004] Jayakrishna Patnaik, Steven Chien, and Athanassios Bladikas, Estimation of bus arrival times using APC data, *Journal of Public Transportation*, vol. 7 (1), 2004, 1-20.

[Penttinen and Piche, 2010] Antti Penttinen and Robert Piche, *Bayesian Methods*. Tampere: Tampere University of Technology, 2010, 7-18.

[Shoshany-Tavory et al., 2014] Sharon Shoshany-Tavory, Ayelet Gal-Tzur, and Niv Eden, Incorporating systems engineering methodologies to increase the transferability of journey planners, *Transportation Research Procedia*, vol. 3, 2014, 631-640.

[Seema and Sheela, 2009] S.R. Seema and Alex Sheela, Dynamic bus arrival time prediction using GPS data, in *Proc. 10th National Conference on Technological Trends*, 2009, 193-197.

[Syrjärinne and Nummenmaa, 2015] Paula Syrjärinne and Jyrki Nummenmaa, Improving usability of open public transportation data, in *Proc. ITS World Congress 2015*, Bordeaux, 2015.

[Syrjärinne et al., 2014] Paula Syrjärinne, Jyrki Nummenmaa, Peter Thanisch, Riitta Kerminen, and Esa Hakulinen, Analyzing traffic fluency from bus data, in *Proc. 10thITS European Congress*, Helsinki, 2014.

[Syrjärinne et al., 2015] Paula Syrjärinne, Peter Thanisch, Jyrki Nummenmaa, Elena Betekhtina, Tero Piirainen, and Juha Lundan, Data based bus schedules, in *Proc. ITS World Congress 2015*, Bordeaux, 2015.

[Tampere Journey Planner] Tampere Public Transport E-service Journey Planner. URL http://reittiopas.tampere.fi [Accessed on January 2015]

[Tampere Open Trip Planner] Tampere Open Trip Planner. URL http://trafficdata.sis.uta.fi:65303/ [Accessed on August 2015]

[Thanisch et al., 2014] Peter Thanisch, Jyrki Nummenmaa, Paula Syrjärinne, Esa Hakulinen, and Riitta Kerminen, Risking the public transport connection, in *Proc. 10thITS European Congress*, Helsinki, 2014.

[Tiesyte and Jensen, 2009] Dalia Tiesyte and Christian S. Jensen, Assessing the predictability of scheduled-vehicle travel times, in *Proc.17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, 416-419.

[Tirachini, 2013] Alejandro Tirachini, Bus dwell time: the effect of different fare collection systems, bus floor level and age of passengers, *Transportmetrica A: Transport Science*, vol. 9 (1), 2013, 28-49.

[Uno et al., 2009] Nobuhiro Uno, Fumitaka Kurauchi, Hiroshi Tamura, and Yasunori Lida, Using bus probe data for analysis of travel time variability, *Journal of Intelligent Transportation Systems*, vol. 13 (1), 2009, 2-15.

[Watkins et al., 2011] Kari Edison Watkins, Brian Ferris, Alan Borning, G. Scott Rutherford, and David Layton (2011), Where is my bus? Impact of mobile

real-time information on the perceived and actual wait time of transit riders, *Transportation Research Part A: Policy and Practice*, vol.45, 839-848.

[Yu et al., 2011] Bin Yu, William H.K. Lam, and Mei Lam Tam, Bus arrival time prediction at bus stop with multiple routes, *Transportation Research Part C: Emerging Technologies*, vol. 19, 2011, 1157-1170.